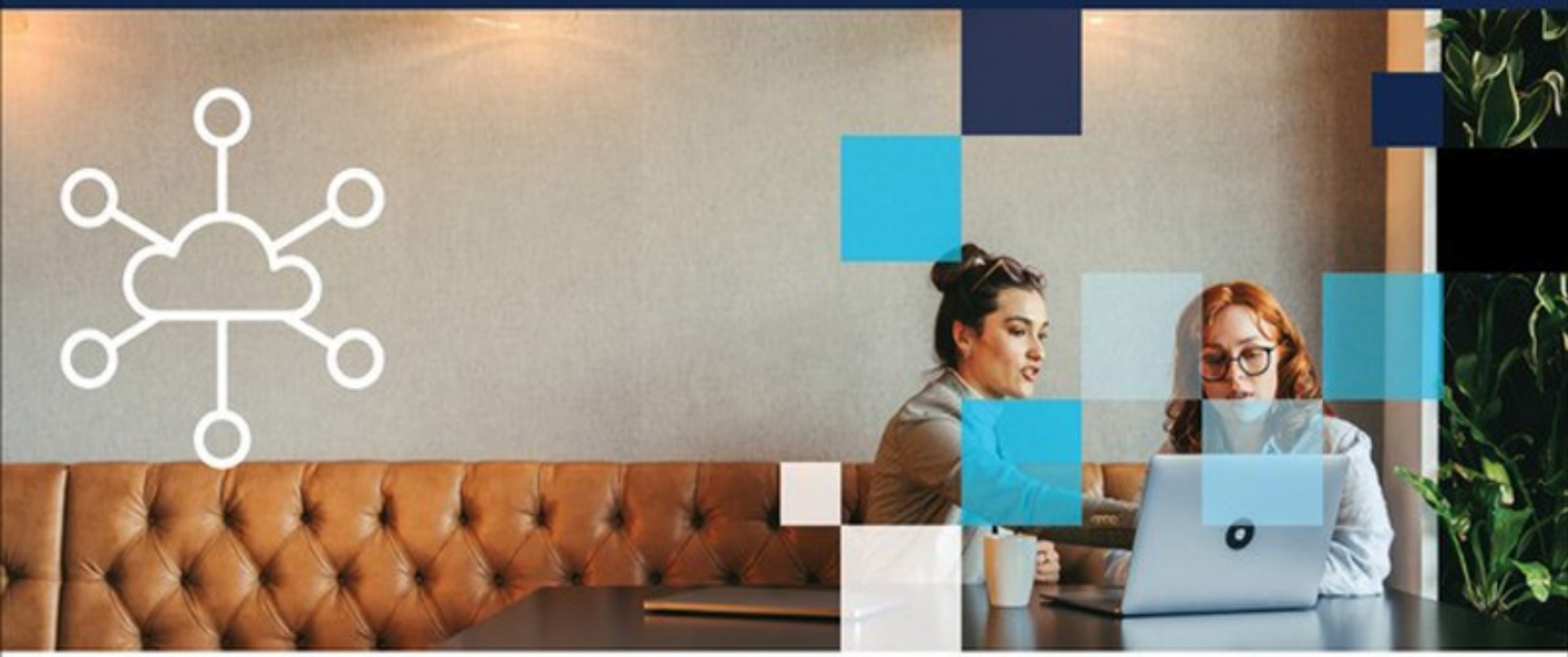
Cisco Catalyst SD-WAN

Design, Deploy and Secure your WAN



ANASTASIYA VOLKOVA
CCIE® X2 (ENT & SEC) NO. 54378

OSVALDO SALAZAR TOVAR

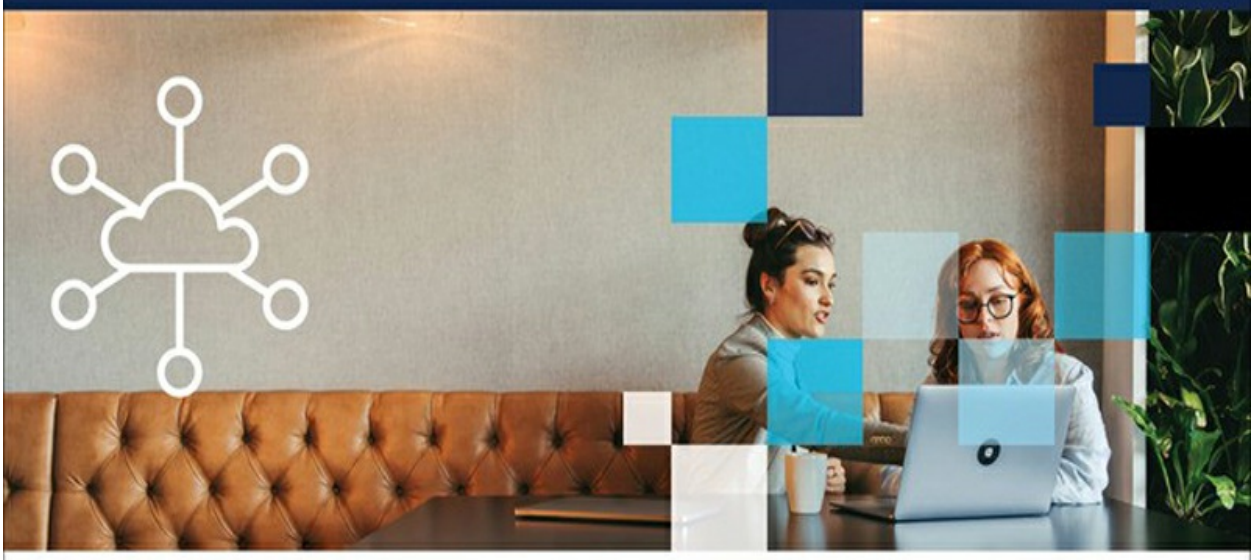
CONSTANTIN MOHOREA
CCIE® X2 (ENT, SEC) NO. 16223, CCDE® NO. 20170054

DUSTIN SCHUEMANN
CCIE® (ENT) NO. 59235

ciscopress.com

Cisco Catalyst SD-WAN

Design, Deploy and Secure your WAN



ANASTASIYA VOLKOVA
CCIE® X2 (ENT & SEC) NO. 54378

OSVALDO SALAZAR TOVAR

CONSTANTIN MOHOREA
CCIE® X2 (ENT, SEC) NO. 16223, CCDE® NO. 20170054

DUSTIN SCHUEMANN
CCIE® (ENT) NO. 59235

ciscopress.com

Cisco Catalyst SD-WAN: Design, Deploy and Secure your WAN/1

Anastasiya Volkova

Osvaldo Tovar

Constantin Mohorea

Dustin Schuemann

Cisco Press

Contents

1 Introduction to Cisco Catalyst SD-WAN

2 Cisco Catalyst SD-WAN Components

3 Cisco SD-WAN Control Plane and Data Plane

4 Onboarding and Provisioning

5 Cisco Catalyst SD-WAN Design and Migration

6 Introduction to Cisco SD-WAN Policies

7 Centralized Control Policies

8 Centralized Data Policies

9 Application-Aware Routing Policies

10 Localized Policies

11 Cisco Catalyst SD-WAN Security

12 Cisco Catalyst SD-WAN Cloud onRamp

13 Cisco Catalyst SD-WAN Programmability

14 Cisco Catalyst SD-WAN Monitoring and Operations

Glossary of Key Terms

Appendix A: Answers to Chapter Review Questions

Table of Contents

Chapter 1. Introduction to Cisco Catalyst SD-WAN

- Transport Independence
- Use Cases Demanding Changes in the WAN
- Cloud Trends and Adoption
- Summary
- Review All Key Topics
- Key Terms
- Chapter Review Questions

Chapter 2. Cisco Catalyst SD-WAN Components

- Data Plane
- Management Plane
- Control Plane
- Orchestration Plane
- Multi-tenancy Options
- Deployment Options
- Summary
- Review All Key Topics
- Key Terms
- Chapter Review Questions
- References

Chapter 3. Cisco SD-WAN Control Plane and Data Plane

Chapter 4. Onboarding and Provisioning

Configuration Templates
Developing and Deploying Templates
Configuration Groups and Feature Profiles
Developing and Deploying Configuration Groups
Onboarding Devices
Key Settings in Device Templates and Configuration Groups
Summary
Review All Key Topics
Key Terms
Chapter Review Questions
References
Chapter 5. Cisco Catalyst SD-WAN Design and Migration
Cisco SD-WAN Design Methodology
Cisco SD-WAN Control Components Design
Cisco SD-WAN Implementation Preparation
Cisco SD-WAN Transport Connectivity
Cisco SD-WAN Data Center Design
Cisco SD-WAN Branch Design
Integrating Cisco SD-WAN with Existing Networks
Summary
Review All Key Topics
Chapter Review Questions
References
Chapter 6. Introduction to Cisco SD-WAN Policies

Chapter 7. Centralized Control Policies
Centralized Control Policy Overview
Use Case 1: Isolating Remote Branches from Each Other
Use Case 2: Enabling Branch-to-Branch Communication Through Data Centers
Use Case 3: Traffic Engineering at Sites with Multiple Routers
Use Case 4: Preferring Regional Data Centers for Internet Access
Use Case 5: Regional Mesh Networks
Use Case 6: Enforcing Security Perimeters with Service Insertion
Use Case 7: Isolating Guest Users from the Corporate WAN
Use Case 8: Creating Different Network Topologies for Each Segment
Use Case 9: Creating Extranets and Access to Shared Services
Summary
Review All Key Topics
Define Key Terms
Chapter Review Questions
References
Chapter 8. Centralized Data Policies
Centralized Data Policy Overview
Use Case 10: Direct Internet Access for Guest Users
Use Case 11: Direct Cloud Access for Trusted Applications
Use Case 12: Application-Based Traffic Engineering
Use Case 13: Protecting Corporate Users with a Secure Internet Gateway

Use Case 14: Protecting Applications from Packet Loss
Summary
Review All Key Topics
Define Key Terms
Chapter Review Questions
References
Chapter 9. Application-Aware Routing Policies
Chapter 10. Localized Policies
Chapter 11. Cisco Catalyst SD-WAN Security
Chapter 12. Cisco Catalyst SD-WAN Cloud onRamp
Chapter 13. Cisco Catalyst SD-WAN Programmability
Cisco Catalyst SD-WAN API Overview
Using the Cisco Catalyst SD-WAN API with Python
Cisco Catalyst SD-WAN Infrastructure as Code
Summary
Review All Key Topics
Key Terms
Chapter Review Questions
References
Chapter 14. Cisco Catalyst SD-WAN Monitoring and Operations
Glossary of Key Terms
Appendix A. Answers to Chapter Review Questions

Chapter 1. Introduction to Cisco Catalyst SD-WAN

This chapter covers the following topics:

• **Introduction to Cisco Catalyst Software-Defined WAN (SD-WAN):** This section examines the benefits and drivers of Cisco Catalyst SD-WAN.

• **Use Cases Demanding Changes in the WAN:** This section covers a variety of use cases that businesses are adopting that are putting pressure on the WAN environment.

• **Cloud Trends and Adoption:** This section explores what is driving businesses to move their business-critical applications to the public cloud and SaaS providers.

Transitioning from a network-centric model to a business intent–based WAN network represents a significant and powerful shift. This change allows for greater simplicity in application deployment and management within the WAN architecture. At the same time, it requires a shift in mindset from focusing on network topology to prioritizing application services. A key challenge for network operations teams is effectively supporting both new and existing applications on the WAN, especially considering their substantial bandwidth consumption and sensitivity to variations in bandwidth quality, such as jitter, loss, and delay.

Improving the WAN environment for these applications is crucial. Moreover,

the increasing demand to migrate to cloud-based applications further exacerbates bandwidth requirements on the WAN. Limited flexibility in connectivity options to accommodate the growing number of cloud applications can lead to costly and challenging provisioning of new applications and services. Many businesses rely on service providers for MPLS L3VPN to manage their WAN routing and network service-level agreements (SLAs). However, this reliance can hinder their ability to adapt to evolving application delivery methods, such as cloud and software as a service (SaaS), in a timely manner. Service providers may take months to implement necessary changes to support these applications, and they may charge exorbitant fees or refuse to make changes altogether. In addition, because service providers control the WAN core, instantiating VPNs independent of the underlying transport becomes extremely challenging, making it difficult, if not impossible, to implement differentiated service levels for individual applications.

Hybrid WANs were originated to address these issues. With a hybrid WAN, businesses acquire additional non-MPLS links and add them to the WAN to provide alternate paths that the applications can take across the WAN environment. Businesses have complete control over these circuits—from routing control to application performance.

Typically, VPN tunnels are created over the top of these circuits to provide secure transport over any type of link. Examples of these types of links are commodity broadband Internet, L2VPN, wireless, and 5G/LTE. This

provides what is called *transport independence*, which makes it possible to use any type of transport underneath the VPN and get deterministic routing and application performance. Some applications can be sent over these commodity links rather than over the traditional service provider–controlled L3VPN MPLS links, and the more latency-sensitive traffic can be sent via the MPLS circuits. A hybrid WAN provides unique granularity of traffic control, redundancy, and resiliency. Figure 1-1 illustrates three common hybrid WAN topologies.

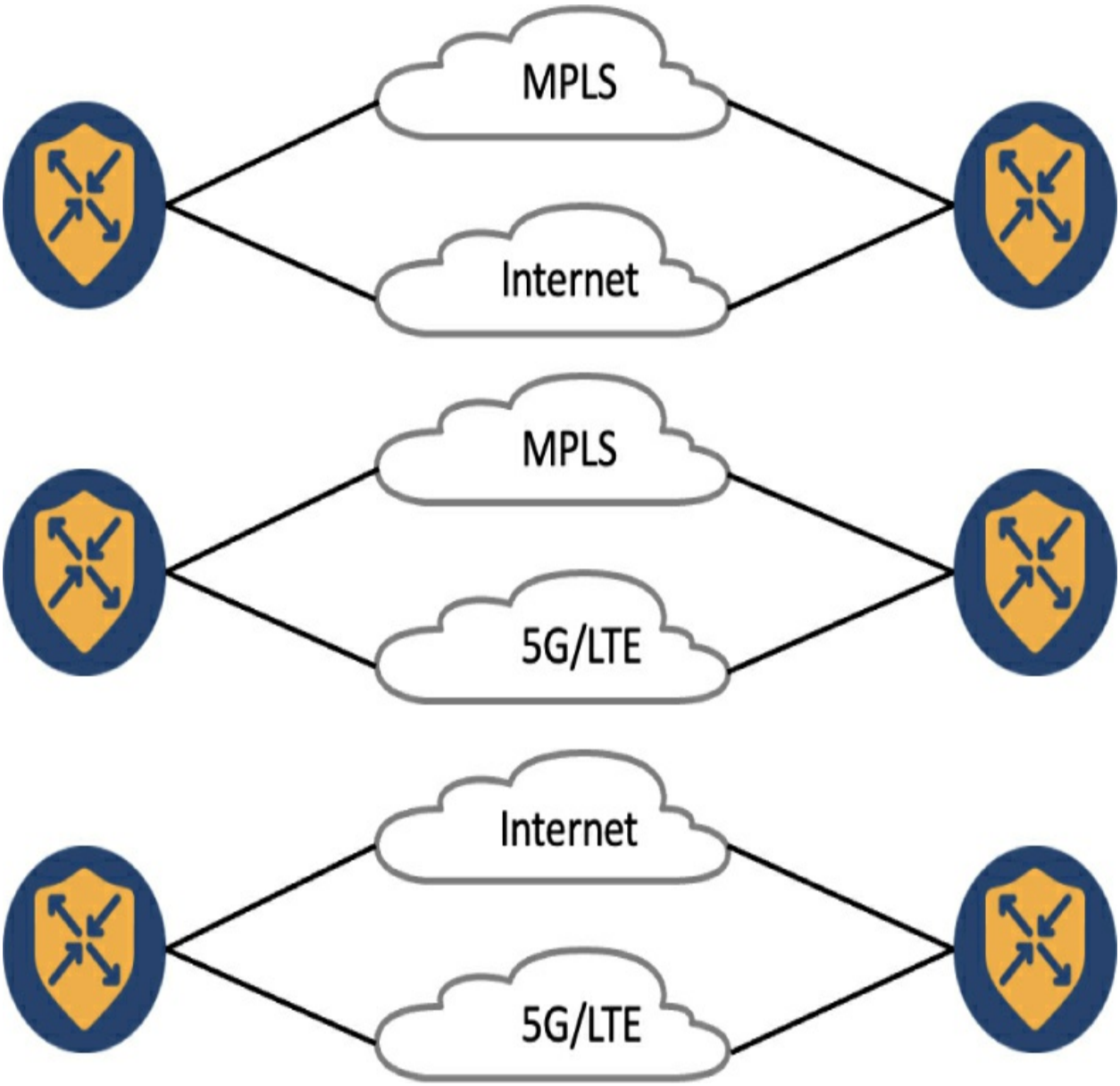


Figure 1-1 *Common Hybrid WAN Topologies*

Hybrid WANs need connectivity that is based on a service topology and can be centrally managed using policies. Currently, WAN connectivity is based on the network topology and managed using a peer-to-peer model. This means routing relationships are established by multiple control planes that operate independently of each other. Routing protocols such as Open Shortest

Path First (OSPF) and Border Gateway Protocol (BGP) are used to establish routing domains across these connections, and IPsec is commonly used to secure the transport. These routing and security control planes run independently of each other and have their own scaling limitations, convergence requirements, and policy enforcement. This means each control plane is required to have its own independent policy and configuration. As a result, when a configuration change is required in the network, it has to be provisioned and propagated across all the control plane peers and for all transports, which creates operational pitfalls and also creates a potential risk of misconfiguration or missing configuration that might cause applications to suffer.



Transport Independence

Cisco Catalyst SD-WAN leverages a transport-independent fabric technology that is used to connect remote locations together. This is accomplished by using overlay technology.

The overlay works by tunneling traffic over any kind of transport between any destination within the WAN environment. This is the VPN concept that was mentioned earlier in this chapter—where, for example, it is possible to connect remote branches, regardless of whether they use MPLS or broadband Internet circuits. Using overlay technology gives true flexibility to routing applications across any portion of the network, regardless of what type of

circuit or transport is in use. This is the definition of transport independence.

With a fabric overlay network, every remote site, regardless of physical or logical separation, is always a single hop away from another. This is of great benefit in terms of application latency and dynamic communication scenarios such as voice or interactive video. It not only provides increased simplicity in terms of network operations but also provides seamless mobility from a user experience perspective. Transport independence is also one of the primary aspects of Cisco Catalyst SD-WAN that allows for the use of flexible, lower-cost commodity circuits rather than high-cost, inflexible static bandwidth. Although service providers can upgrade the bandwidth of a circuit, cost is usually a barrier. In addition, there are many times that, based on the type of circuit the bandwidth is riding on, an entire physical circuit upgrade or swap may be more likely. An example of this is having a 100Mbps MPLS handoff wherein the physical circuit it is delivered on is also only 100Mbps. In cases like this, another higher-speed port on the provider side is required, such as a Gigabit or 10 Gigabit Ethernet port.

Many times, the circuit may ride over a different type of medium, and the entire circuit and delivery mechanism must be changed—for example, trying to go from a 45 Mbps DS3 to a Gigabit Ethernet link. All of this takes time—sometimes a considerable amount of time—and it is one of the things SD-WAN was created to address. A business can typically order a high-speed commodity Internet circuit and have it delivered within weeks. The new Internet circuit can be immediately added to the environment, and the

business can take advantage of it by using SD-WAN. There are situations where multiple branch locations need to act as a single large branch across the WAN. This means having a virtual fabric over disparate transports such as MPLS and the Internet.

Given everything that has been covered thus far, it is important to now look at an example of a Cisco Catalyst SD-WAN diagram. Figure 1-2 provides a high-level overview of a Cisco Catalyst SD-WAN environment and how users, devices, and applications fit into the overall design.

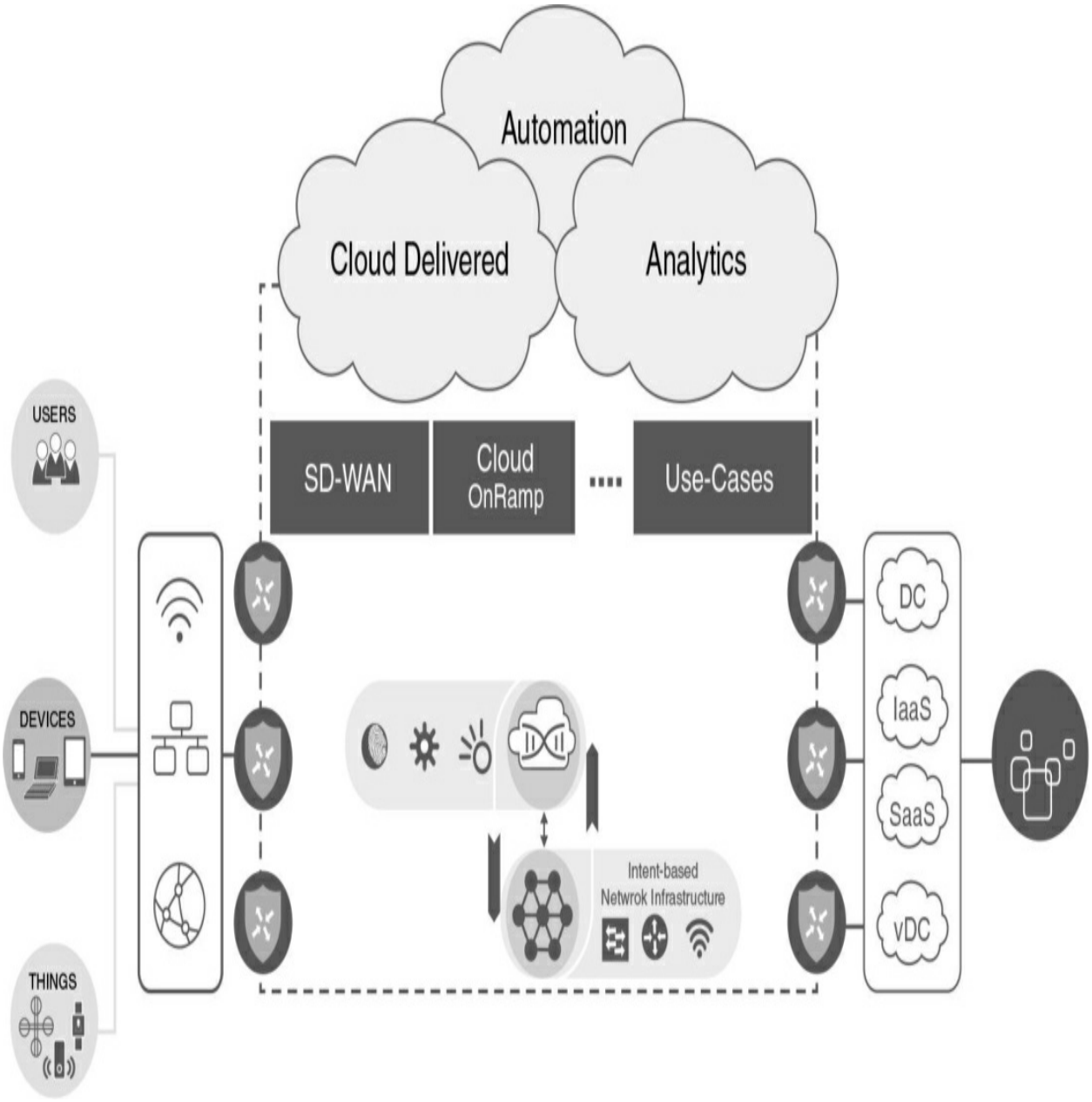


Figure 1-2 *High-Level SD-WAN Overview*

Moving from a network-centric WAN to an application- and services-focused WAN requires a different view of the WAN. Figure 1-3 illustrates a business intent–based network, its components, and how they fit within the new model.

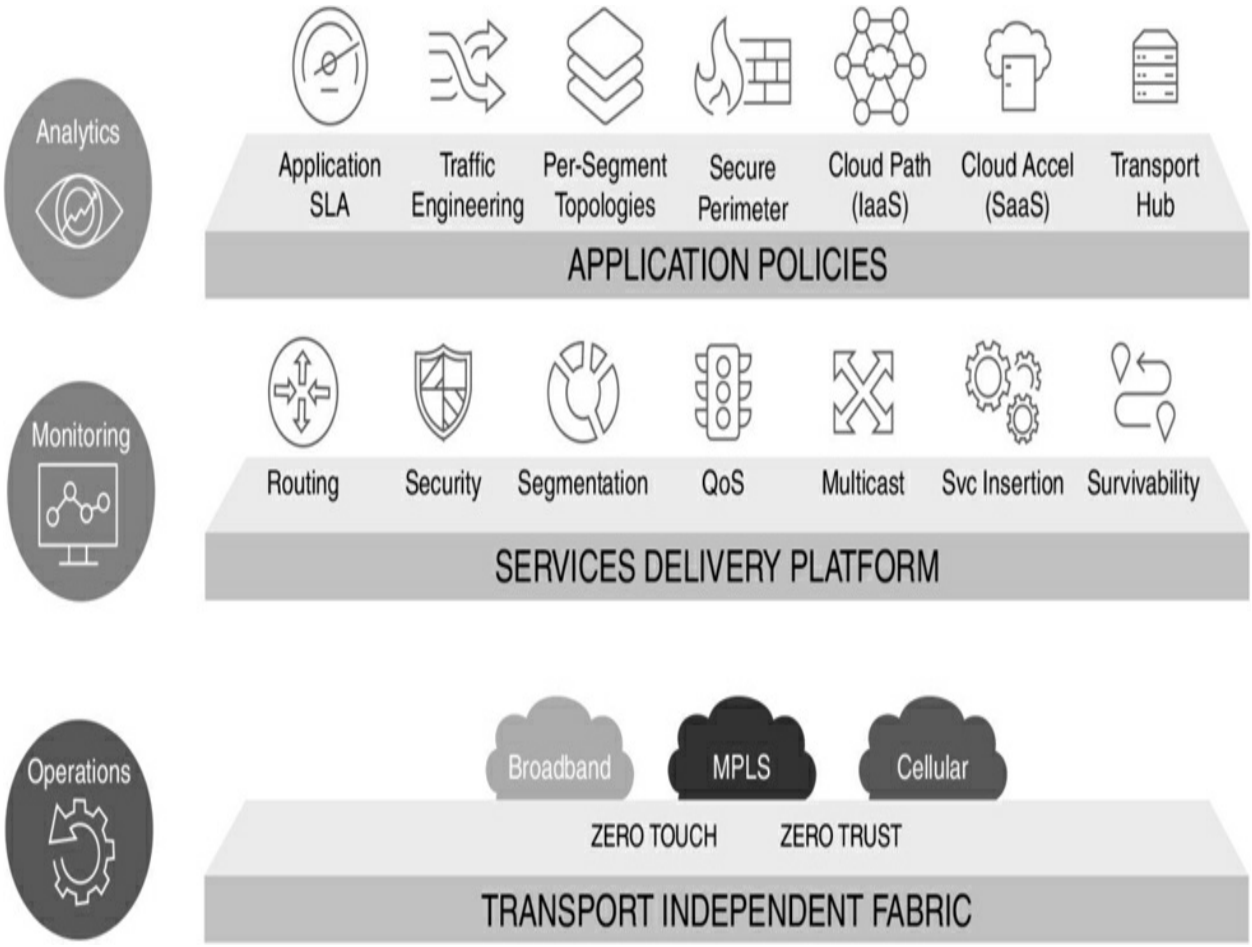


Figure 1-3 *Business Intent–Based Network Components*

Rethinking the WAN

If we were to redefine the current WAN technology and approach, we would want to make some fundamental changes to how WANs are constructed and managed today to address modern business and application requirements. These changes would involve the following key areas:

• Secure elastic connectivity

• Cloud-first approach

• Application quality of experience

• Agile operations

From a security perspective, end-to-end segmentation and policy are critical. The control, data, and management planes must be separated across the entire environment. The environment should be able to support native encryption that is robust and scalable, and it should offer lightweight key management and leverage a zero-trust model, meaning every aspect of the onboarding process must be authenticated and verified.

Rethinking the WAN from a connectivity perspective, these elements would be built on top of security functionality by integrating routing, security, and policy for optimal use of connectivity. The solution must allow for multiple types of transport connectivity options simultaneously and ultimately create a transport-independent operation model. Scalability, both horizontally and vertically, is necessary at any layer. In addition, advanced VPN capabilities and topologies to address any business intent or requirements are critical.

In terms of application support, the solution should support full application awareness across all elements in the system and offer built-in optimization techniques for the networks and applications. The network has evolved to be application aware, and it must be capable of choosing the most optimal path to connect to on-premises or cloud-based applications. The application experience must be optimal in terms of both access and security.

When it comes to the operation of this new application- and services-oriented WAN, network operations staff must be able to define networkwide policies that leverage templates rather than just use a device- or node-level policy. The controller must have the ability to coordinate the paths between the WAN edge routers, based on centralized policy orchestration. As organizations' network requirements change and evolve over time, the policy should be able to be changed in a single place. This not only reduces the amount of time spent on configuration but lowers the risk associated with misconfiguration errors. Programmable, open application programming interfaces (APIs) should be available to provide northbound access for automation and orchestration capabilities. Support for southbound APIs for integration with other solutions should also be included.

Use Cases Demanding Changes in the WAN

Currently there are many reasons to look at enhancing the WAN environment —from load balancing traffic to ensuring that applications have the best performance possible. The following sections cover some of the use cases that are causing changes to the WAN.

Bandwidth Aggregation and Application Load Balancing

There are many different use cases that demand changes to traditional WAN architectures. Some are as simple as businesses wanting bandwidth aggregation, which is the ability to use both public and private transports together at the same time. For example, an organization might use A + B

rather than A or B, meaning the secondary transport link (Link B) usually sits idle without any traffic using it until Link A fails. However, in a hybrid WAN approach, being able to leverage multiple links at the same time provides the ability to use bandwidth from both links. This is considered an A + A, or active/active, scenario.

Application load balancing is achieved using these types of designs as well. A hybrid environment featuring application load balancing allows for greater application performance at a fraction of the cost of two premium transport links. It also increases scalability and flexibility without any security compromise. Figure 1-4 illustrates the various options of application load balancing over multiple links in a hybrid environment. You can see that, by default, per-session active/active load sharing is achieved. Weighted per-session round-robin is also configurable on a device basis. Application pinning, or forcing an application to take a specific transport, is also something that can be enforced via policy. Similarly, application-aware routing or SLA-compliant routing is achieved by enforcing a policy that looks for specific traffic characteristics, such as jitter, loss, and delay, to determine the path the application should take over the available transports.

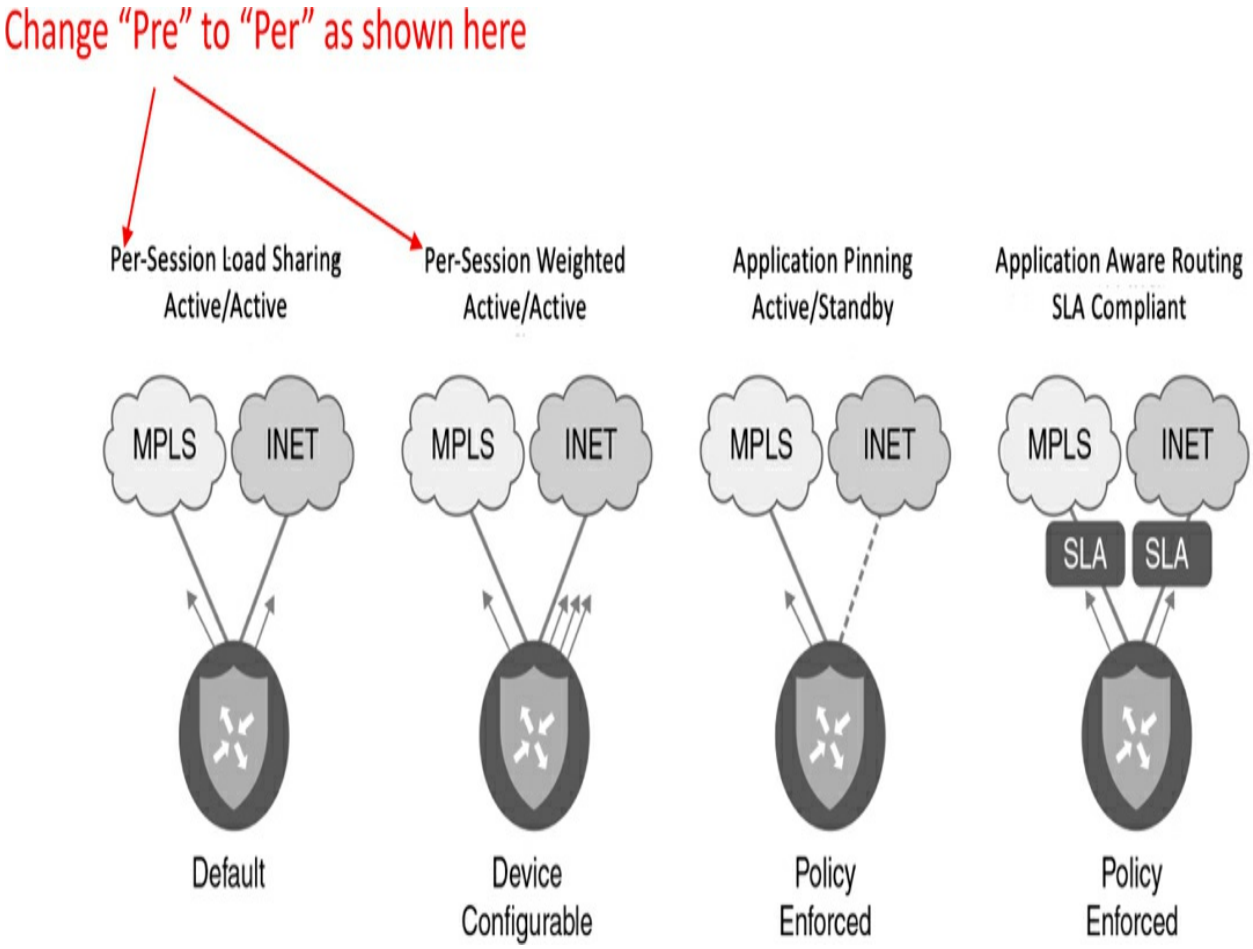


Figure 1-4 *Application Load-Balancing Options*



Protecting Critical Applications with SLAs

The requirement and capability to provide SLA for critical applications is another use case that drives changes in the WAN. This is accomplished by being able to route traffic based on the application requirements, as mentioned briefly earlier. This type of routing provides statistics on how the applications are performing. An SLA determines whether an application is adhering to the policy that has been created and whether it is performing

properly. In the application is experiencing some sort of impairment, such as jitter, loss, or delay, the application can be routed to another transport which will ensure that the application is within policy and able to perform to the SLA. Figure 1-5 illustrates this scenario.

A good example of providing SLA for critical applications in a hybrid WAN environment would be an MPLS link and an Internet link. If the MPLS link is experiencing >2% packet loss and the Internet link is not, it might be appropriate to route the application over the Internet link to ensure that the application is functioning properly and users are having the best possible experience interacting with the application. In the example in Figure 1-5, the policy says that if the application is experiencing any jitter greater than or equal to 10 ms, then it should prefer another transport. In this example, Path 2 and Path 3 are above the threshold, so Path 1 will be preferred.

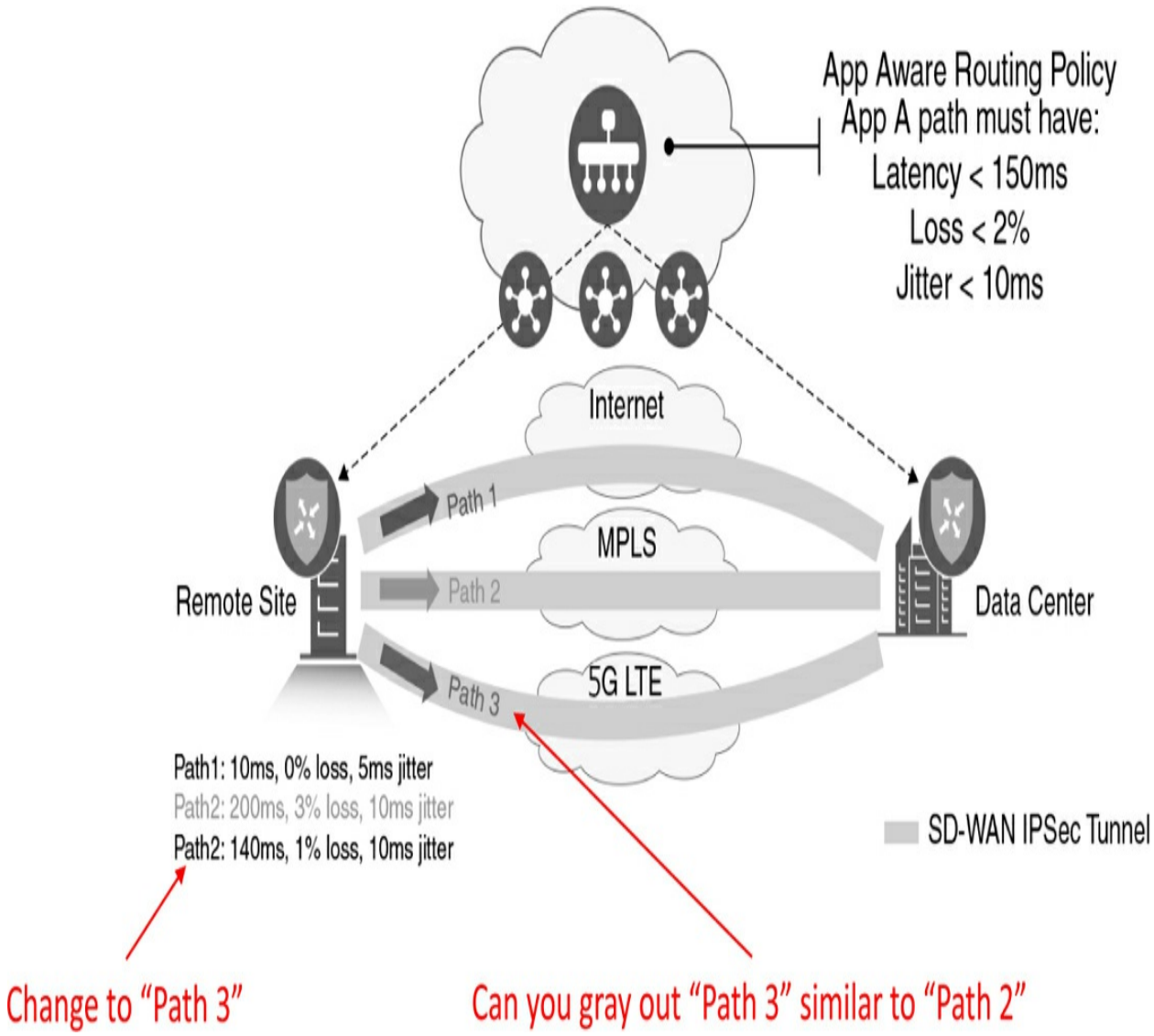


Figure 1-5 *Routing Based on Application Performance*

End-to-End Segmentation

Segmentation is another use case that drives changes in the WAN. Oftentimes, businesses face different corporate or regulatory requirements that require separation. For example, research and development may need to be segmented from the production environment. There may be extranets that connect to partners, or the business may be merging or acquiring another

business; the networks may need to be able to communicate, but segmentation may still be required between the two. This type of situation might require multiple topologies that can be managed as one. Figure 1-6 illustrates an end-to-end segmentation topology, along with how different VPNs are carried over the tunnels. Each of these tunnels terminates at an edge router within the environment.

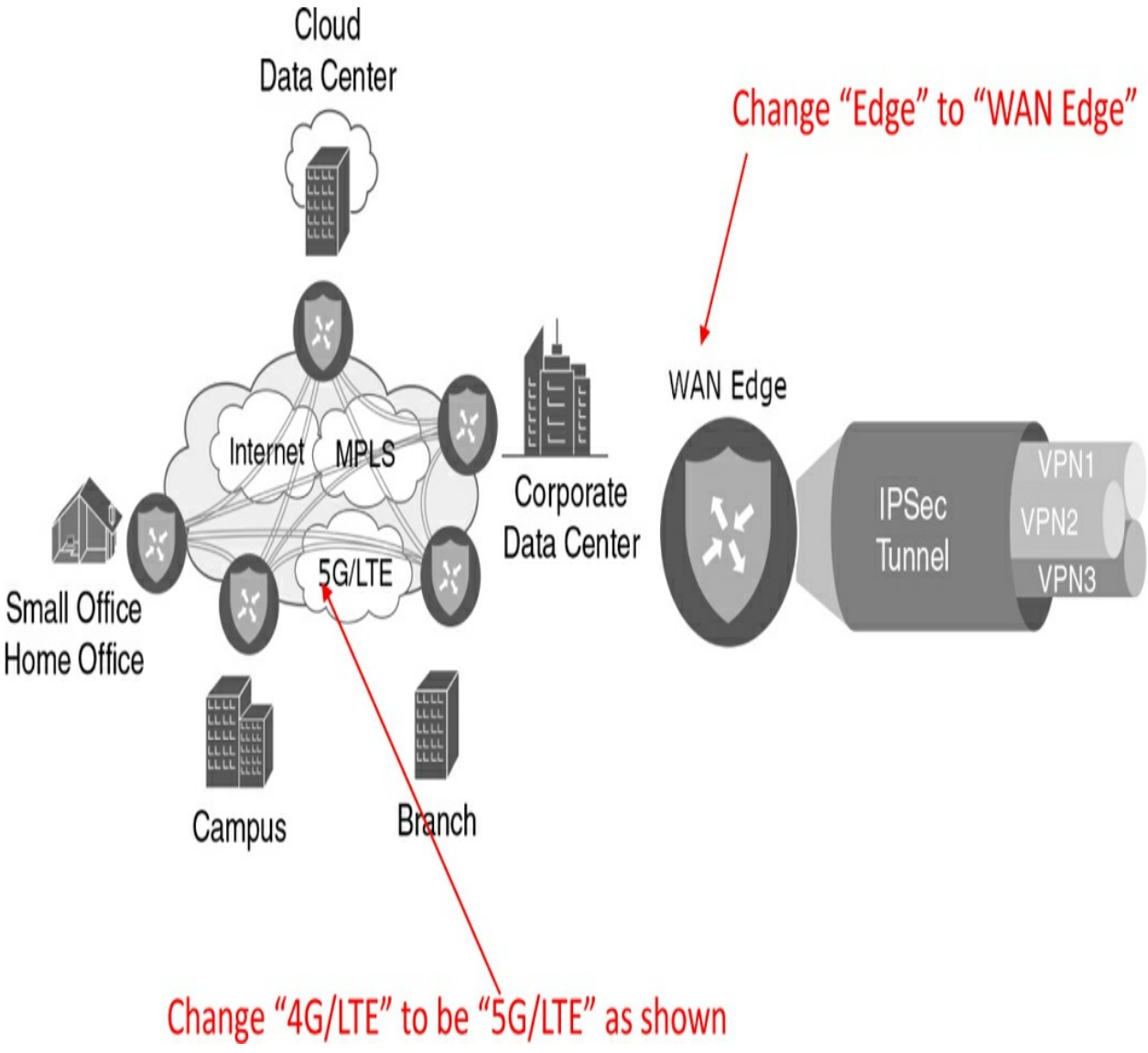


Figure 1-6 *End-to-End Segmentation*



Direct Internet Access

Direct Internet Access (DIA) gives branches the capability to send traffic directly out the local Internet transport instead of carrying it all the way back to a centralized data center to be inspected. This allows for cloud-based applications to go directly to the Internet and cloud service providers without having to use unnecessary WAN bandwidth. This method is increasingly being adopted.

Figure 1-7 illustrates the traditional way that cloud applications are accessed, which causes suboptimal performance for users trying to access these applications. As mentioned earlier, this method also puts a strain on the WAN infrastructure, as the expensive and limited WAN bandwidth is being consumed by applications that could be sent directly to the Internet from the remote site. It also introduces increased application latency, as the traffic has to cross the entire network to get to the data center to reach the Internet.

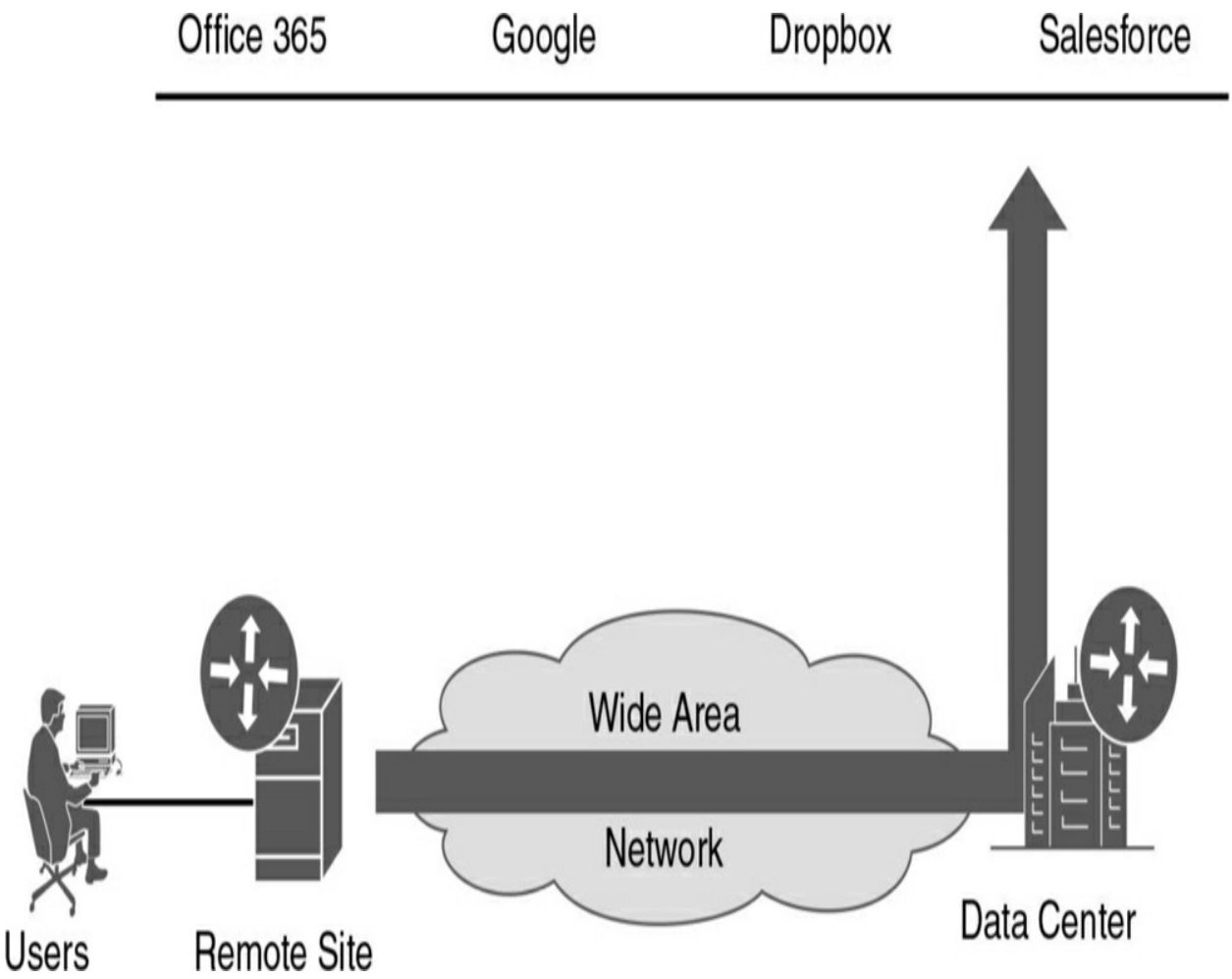


Figure 1-7 *Traditional Cloud Application Access via WAN*

Redesigning the WAN allows for different mechanisms that will allow for better performance and scalability. A great example of this is using the Direct Internet Access design to offload latency-sensitive cloud applications directly to the Internet. This method also provides the flexibility of allowing a local firewall or inspection device in the branch to ensure that the branch is protected from any malicious threats coming into the local branch Internet link. Figure 1-8 shows an example of what this would look like in a new WAN environment.

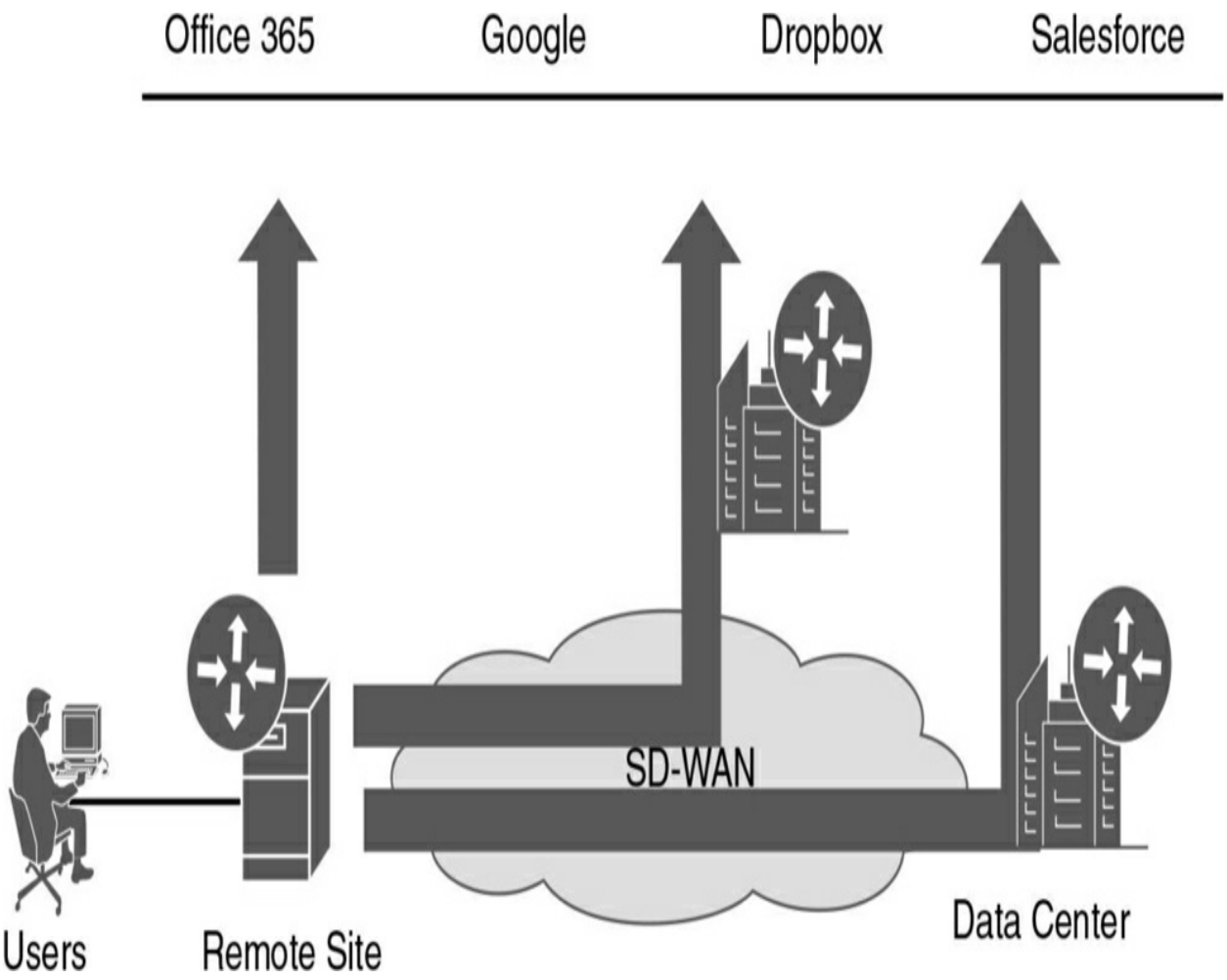


Figure 1-8 *Direct Internet Access and Cloud Access Topologies*

Fully Managed Network Solution

Finally, there is a use case that allows for the business to simply let someone else, such as Cisco or a Cisco partner, manage the network as a fully managed solution. This option provides the flexibility to have the network managed as a whole for the business and also allows the business to have control over the policy and reporting portion of the managed service. This is becoming a more attractive option for customers that want to move to an OpEx model. They can pay for their network on a subscription basis rather

than use the traditional CapEx model and is analogous to paying an electric or cell phone bill. The consumption models available today are really opening up new options for customers.

Note

All of these use cases and technologies are covered in detail in this book.



Cloud Trends and Adoption

Cloud adoption has been taking the industry by storm. Over the years, the reliance on the cloud has grown significantly, starting with music, movies, and storage and moving into software as a service (SaaS) and infrastructure as a service (IaaS). Today, there are many aspects of businesses—such as application development, quality assurance, and production—that are running in the cloud. To make things even more complicated, companies are relying on multiple cloud vendors to operate their businesses.

Companies are also struggling with things such as shadow IT and backdoor applications in their environments. Lines of business are going to cloud providers on their own, without any knowledge or guidance from IT departments, and spinning up applications on demand in the cloud. As companies migrate workloads to the cloud, they are typically looking to keep

the same set of policies they use on premises when carrying traffic into the cloud. Security and privacy are both concerns, and the potential loss of confidential information or intellectual property could damage an organization’s brand and reputation. The risks are significant.

Furthermore, the applications in the cloud, whether legitimate production or development applications, still require certain levels of priority and treatment to ensure that they are being delivered properly to the users who consume them. Cisco Catalyst SD-WAN can help ensure that the applications are being treated appropriately and that the experience for the users is adequate.

Figure 1-9 illustrates the demand on the WAN and how the Internet is becoming critical to the operations of the business.

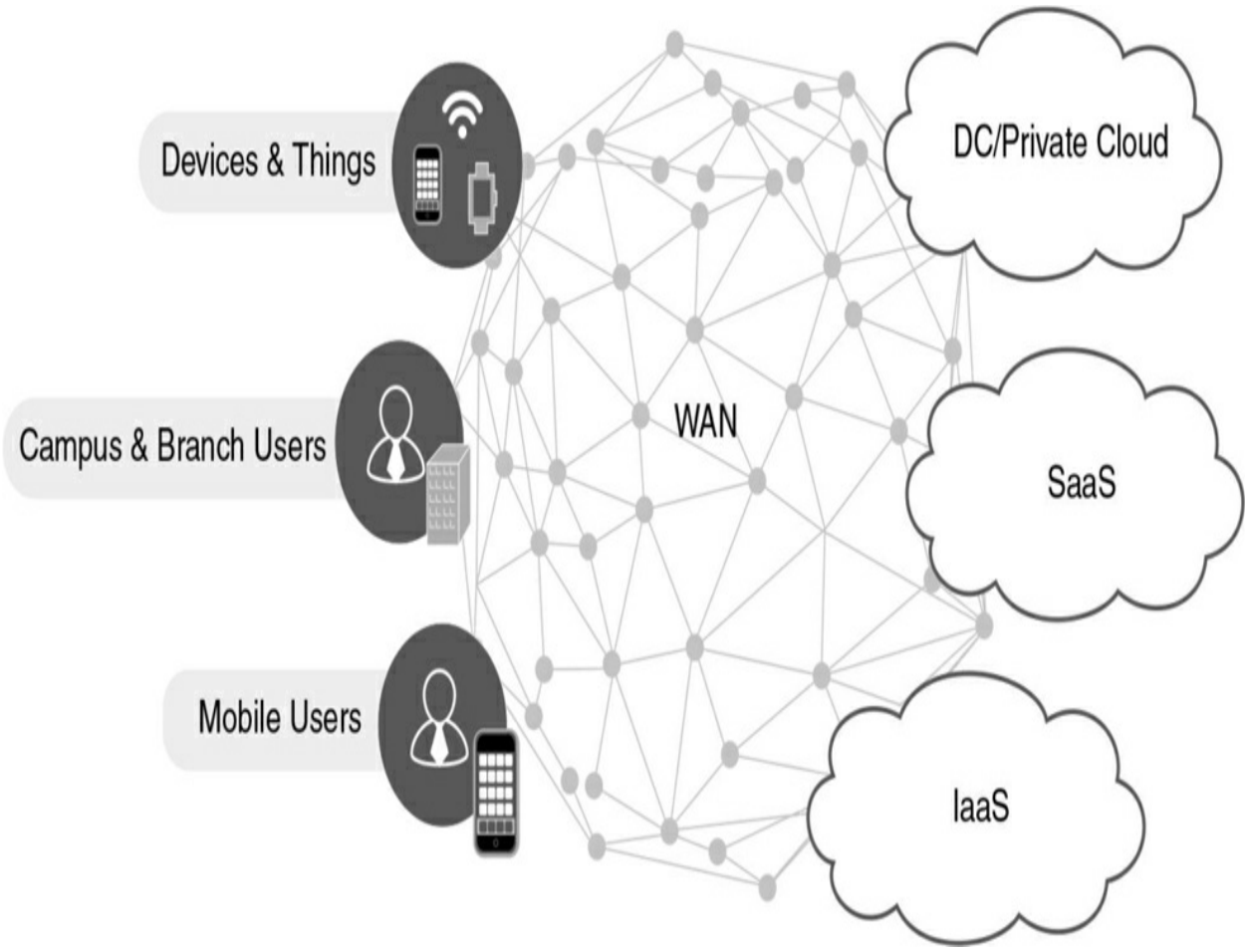


Figure 1-9 *Demand on the WAN for Internet-Based Applications*

Direct Internet Access can assist with this, as mentioned earlier. By being able to detect application performance through one or more Direct Internet Access circuits, the edge routers are able to choose the best-performing path based on the application-specific parameters. If one of the links to the cloud application fails or experiences performance degradation, the application can automatically fail over to another Direct Internet Access link. This process is fully automated and requires no interaction from the network operations staff. Figure 1-10 shows this scenario with multiple Direct Internet Access links.

Technet24

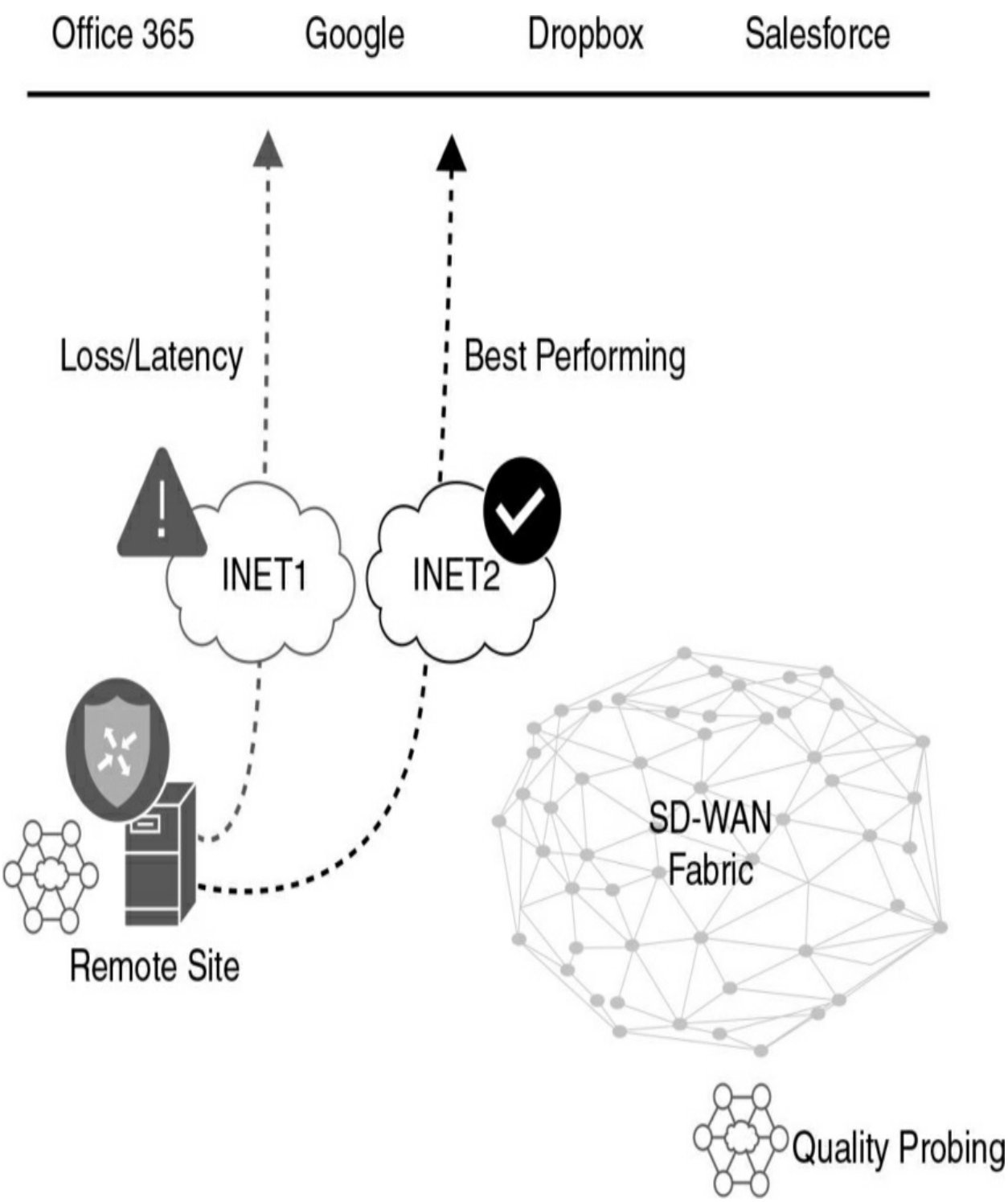


Figure 1-10 *Multiple Direct Internet Access Links to Cloud Applications*

This concept also works in an environment that has a remote branch site that

has a local directly attached Internet link as well as an Internet link within a centralized data center. The same process takes place: The application performance is measured and the path that provides the best performance is chosen for the application. Similarly, Direct Internet Access can protect against blackout or link failures because of the redundancy built into the solution via the multiple available paths. Figure 1-11 illustrates a scenario with a local directly attached Internet link and an Internet link available in a centralized data center. Again, this leaves the router to make the decision based on the policy and application parameters configured. These decisions are fully automated and made on a per-application and per-VPN basis, which ultimately provides an amazing amount of flexibility and control over the application performance within the environment.

Technet24

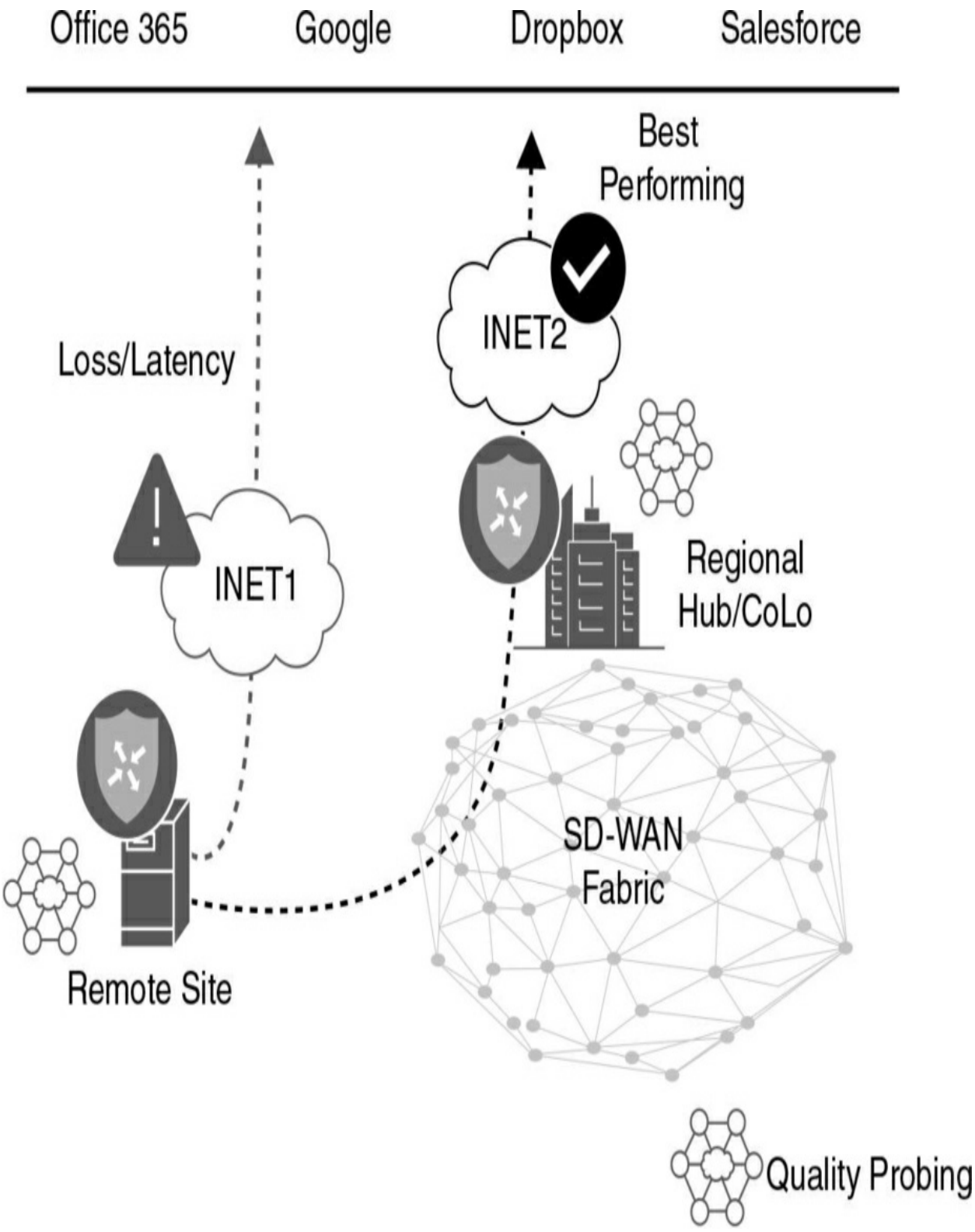


Figure 1-11 *Direct Internet Access and Centralized Internet Link to Cloud Applications*

There is one more force driving organizations to the cloud that we need to mention: As companies are moving their custom-developed applications to public cloud providers such as Amazon Web Services, Microsoft Azure, and Google Compute, the companies want to connect these cloud providers to their WANs. By connecting these cloud providers to your WAN, you can continue to maintain a common routing policy while also enforcing your businesses security policy. In some cases, you can now leverage a cloud provider's network for transport for your WAN sites; this is referred to as middle-mile optimization. By taking advantage of middle-mile optimization, you can provider a simpler, more efficient, and scalable solution, using the cloud provider's global infrastructure.

Summary

This chapter provides a high-level overview of network challenges for businesses and their operations staff. It also covers common business and IT trends the industry is seeing and how they impact networks today. Organizations and their IT staff are rethinking the WAN environment in order to attain certain benefits. Cloud applications and the influx of large amounts of within the network are causing strain on the WAN. Businesses are looking at ways to alleviate the pressure being put on the WAN—and on the organization as a whole. The use cases introduced in this chapter are all covered in depth in the upcoming chapters in this book.

As you have seen, cost is not the only factor driving organizations to look at

Technet24

SD-WAN. Application performance, security, segmentation, user experience, redundancy, and resiliency are also key drivers that point to SD-WAN.

Review All Key Topics

Review the most important topics in this chapter, noted with the Key Topic icon in the outer margin of the page. Table 1-1 lists these key topics and the page number on which each is found.



Table 1-1 *Key Topics for Chapter 1*

Key Topic Element	Description	Page Number
Section	Transport independence	
Section	Protecting critical applications with SLAs	
Paragraph	Cisco Direct Internet Access	

Key Terms

Define the following key terms from this chapter and check your answers in the glossary:

application programming interface (API)

Cisco Software-Defined WAN (Cisco SD-WAN)

cloud

infrastructure as a service (IaaS)

service-level agreement (SLA)

software as a service (SaaS)

Chapter Review Questions

1. What are some benefits businesses are looking for from their WAN? (Choose three.)

a. Lower operational complexity

b. Increased usable bandwidth

c. Reduced uptime in branch locations

d. Topology dependence

e. Improved overall user experience

2. What are some of the tools or technologies that may need to be

Technet24

implemented when redundant links are used in branch locations? (Choose three.)

a. Administrative distance

b. Traffic engineering

c. Redistribution

d. Loop prevention

e. Preferred path selection

3. For what reasons are organizations looking to deploy SD-WAN? (Choose two.)

a. To take all routing control from the service provider

b. To create end-to-end SLAs for the organization’s traffic

c. To offload all routing control to the service provider

d. To leverage the service provider’s SLA for end-to-end traffic

4. What are some of the benefits of SD-WAN? (Choose four.)

a. Lower cost

b. Improved user experience

c. Transport independence

d. Increased cloud consumption

e. IoT devices

f. Increased bandwidth

5. What are some of the transport options for SD-WAN? (Choose three.)

a. Dual MPLS

b. Hybrid WAN

c. Dual route processor

d. Hybrid single link

e. Dual Internet

6. True or false: Direct Internet Access is used to offload applications directly to the data center.

a. True

b. False

Technet24

7. What are two of the driving forces for cloud adoption? (Choose two.)

a. Middle-mile optimization

b. Cloud application migration

c. Increased costs

d. Increased complexity

Chapter 2. Cisco Catalyst SD-WAN Components

This chapter covers the following topics:

• **Data Plane:** This section discusses the physical and virtual routers that actually carry data traffic.

• **Management Plane:** This section introduces the component that handles most of the day-to-day tasks in managing the Cisco Catalyst SD-WAN fabric.

• **Control Plane:** This section covers the component that handles all policies and routing.

• **Orchestration Plane:** This section introduces the component that facilitates discovery, authentication, and facilitation of the fabric.

• **Multi-tenancy Options:** This section introduces the various multi-tenancy options available in Cisco Catalyst SD-WAN.

• **Deployment Options:** This section covers the various deployment options, including Cisco cloud, private cloud, and on-premises deployments.

This chapter introduces the various components that make up the Cisco Catalyst SD-WAN architecture as well as the various deployment options. At a high level, these components can be grouped based on the purpose they play in Cisco Catalyst SD-WAN:

Technet24

• Data plane

• Management plane

• Control plane

• Orchestration plane

In traditional networks, the management plane, data plane, and control plane are all on the same router, and together they facilitate communication within the network. A traditional router has network interfaces and line cards (which handle forwarding of data packets); this is a data plane. A CPU module, which handles calculating a routing table and advertising networks to the rest of the network is a control plane, and the command-line interface (CLI) that is used to configure the router is a management plane. At the CLI, you type commands, and those commands program the CPU and line cards to act on your intent. Each router in a network has these three components.

A traditional network has a number of routers, each of which needs to be programmed independently to achieve the desired operational state of the network. As networks get larger, the amount of human intervention required to configure the environment dramatically increases, potentially creating complexity. Each router must calculate its own routing table from its perspective of the network. For example, suppose you have a network with 6000 routes. Whenever there is a change in the network, each router may potentially have to process routing updates for each of these routes. This

means the router must have the available CPU and memory required to process these updates, and this creates a lot of overhead. Tuning the routing table on a network with a large number of sites and routes—whether the network is full mesh, hub and spoke, partial mesh, and so on—can quickly become very complex. In addition, because each router is programmed individually, when you program the network on a router-by-router basis, you run the risk of undesired results due to improper design or human error on the CLI.

Cisco Catalyst SD-WAN is a distributed architecture that provides a clear separation between the management plane, control plane, and data plane. Figure 2-1 illustrates how the components fit into the architecture.

Technet24

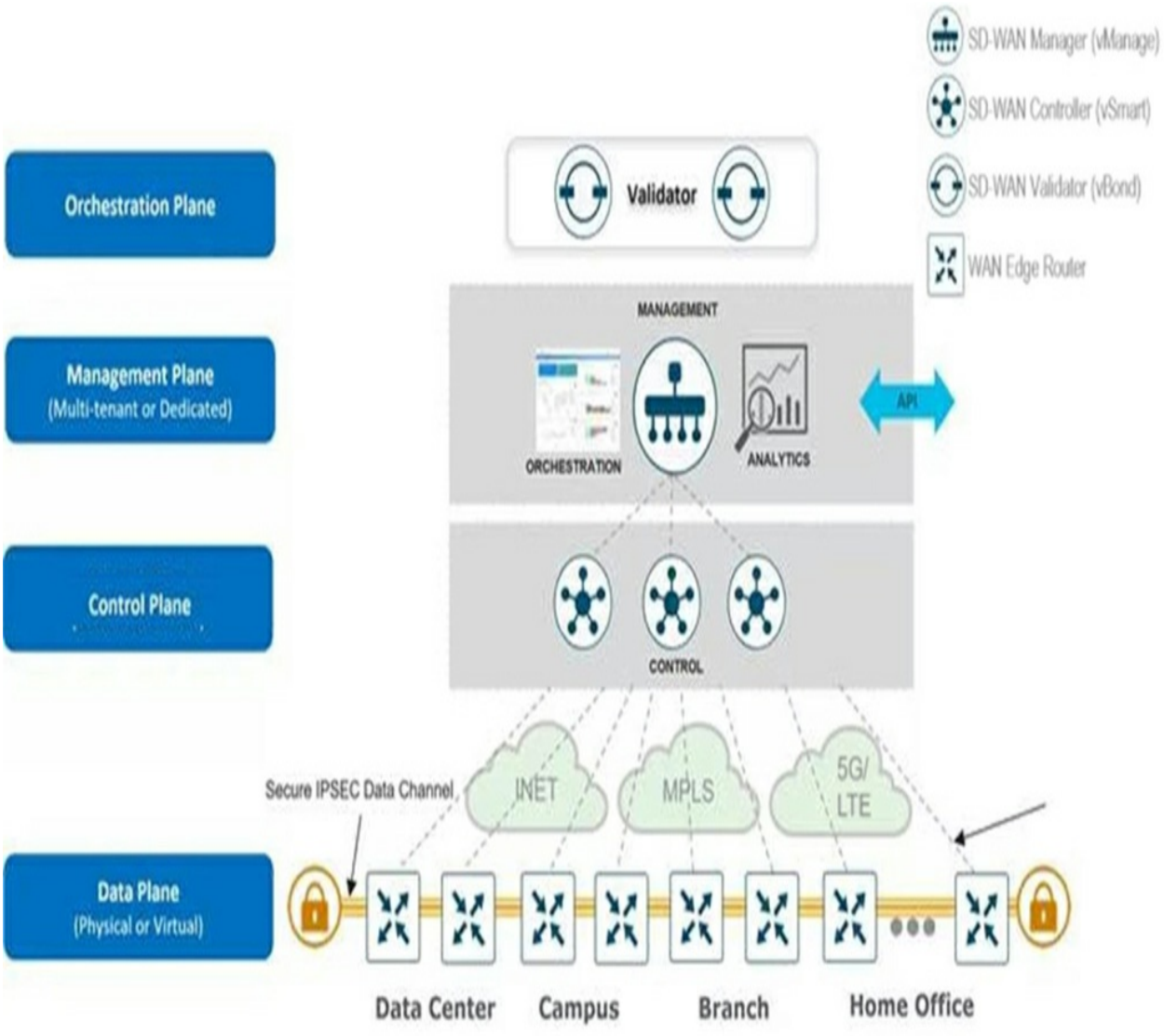


Figure 2-1 *Cisco Catalyst SD-WAN Distributed Architecture*



The Cisco Catalyst SD-WAN distributed architecture differs from traditional network architectures in that it allows you to support large-scale networks while reducing operational and computational overhead. Catalyst SD-WAN separates the data plane, the control plane, and the management plane from each other.

Because the control plane knows about all routes and nodes on the network, you have to calculate the routing table only once and can distribute the information to all the necessary nodes as a single routing update rather than have every router send routing updates to the others, with each determining its own Routing Information Base (RIB). This greatly reduces the overhead on the network and enables you to reduce required resources on the routers so that you can bring additional features and capabilities to your edge devices. Because you have a complete view of the network, you can create a common network policy across the entire SD-WAN fabric—and the management plane needs to program it only once. As new devices are added to the network, they receive the same policy as well, ensuring that the network is operating as expected. This book shows how you can create various topologies and policies with ease while increasing scale and capability.

Note

In 2023 Cisco SD-WAN was rebranded as Cisco Catalyst SD-WAN. In addition, starting with Cisco Catalyst SD-WAN Release 20.12, the following component changes apply:

- Cisco vManage is now Cisco Catalyst SD-WAN Manager.
- Cisco vAnalytics is now Cisco Catalyst SD-WAN Analytics.
- Cisco vBond is now Cisco Catalyst SD-WAN Validator.

Technet24

• Cisco vSmart is now Cisco Catalyst SD-WAN Controller.

• Cisco SD-WAN Controllers (vManage, vSmart, and vBond together) are now Cisco Catalyst SD-WAN Control Components.

This change is reflected in the GUI and the official documentation. However, because the older names are more familiar to customers and are used in the ENSDWI certification exam, this book uses the new names but reminds you of the old names where appropriate.



Data Plane

Traditionally, the data plane has been composed of the physical interfaces that the physical layer plugs into (for example, Ethernet, fiber, serial). As mentioned previously, this is analogous to the line cards on routers and switches. In Cisco Catalyst SD-WAN, the data plane consists of WAN Edge devices, which could be Cisco IOS XE SD-WAN routers or legacy Cisco vEdge routers. Data plane devices may be deployed at branches, data centers, large campuses, colocation facilities, or in the cloud. At each site, you can have a single WAN Edge router or multiple WAN Edge routers, depending on your redundancy requirements.

The data plane is where the SD-WAN overlay resides and is the layer that forwards user, server, and other network traffic. Both IPv4 and IPv6 are supported for transport within the data plane. In addition, data policies (such as QoS and Application-Aware Routing) are enforced within the data plane.

Each WAN Edge router forms data plane connections to other WAN Edge routers within the SD-WAN overlay for the purposes of transporting user traffic. Data plane connections are only established between data plane devices. These tunnels are typically secured via Internet Protocol Security (IPsec). As described previously, the data plane has native segmentation. The segmentation information is encapsulated as defined in RFC 4023 and is carried across the SD-WAN overlay. Segmentation allows the network administrator to build separate instances of the data plane, depending on business requirements and regulations. The original data packets are typically encapsulated with IPsec, providing encryption and authentication.

Cisco Catalyst SD-WAN supports GRE as another method of data encapsulation. GRE provides less overhead but lacks all the security that IPsec provides, and it is used less often. Throughout this book, you can assume that IPsec encapsulation is being used unless noted otherwise.

Figure 2-2 illustrates the Cisco Catalyst SD-WAN packet structure.

Technet24

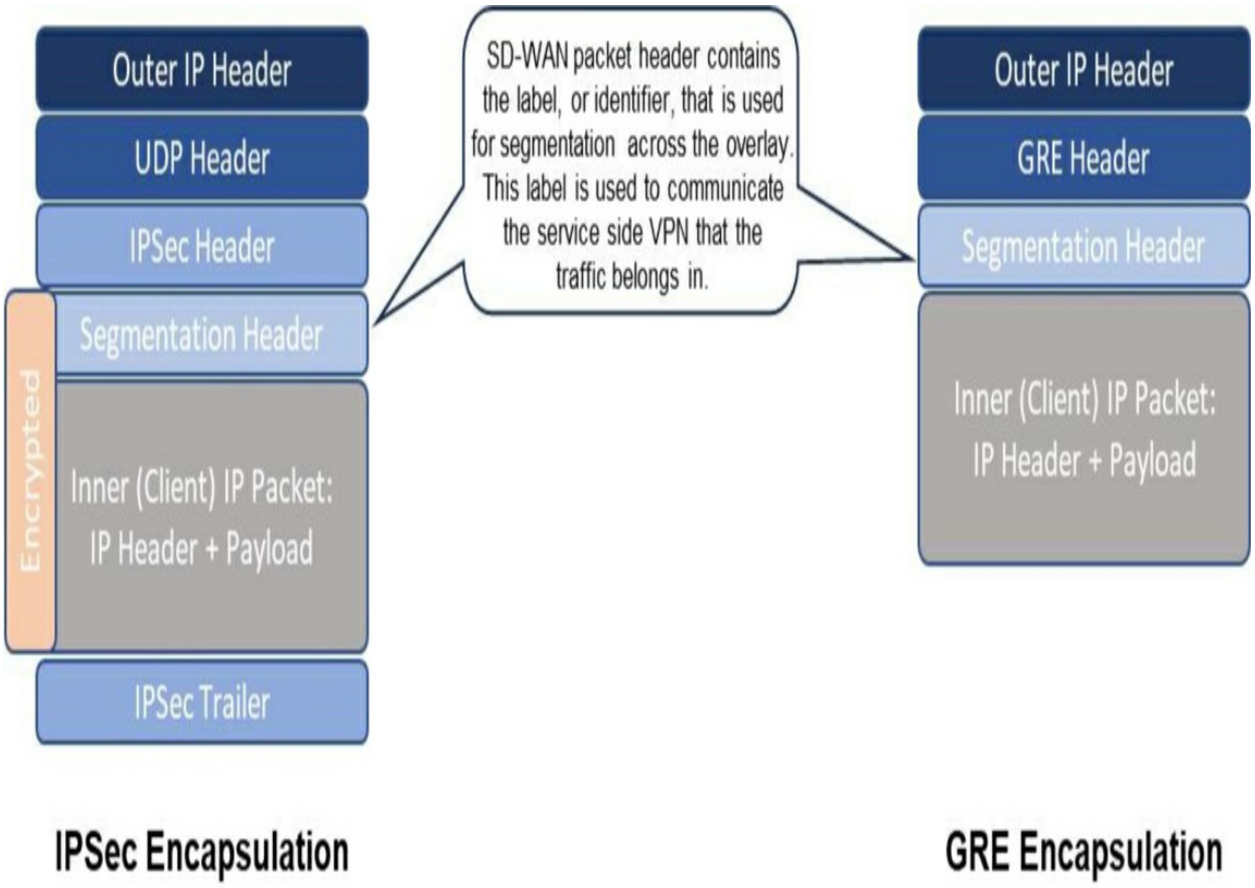


Figure 2-2 *Cisco Catalyst SD-WAN Packet Format*

Cisco Catalyst SD-WAN can support diverse topologies that are unique to each VPN segment or data plane instantiation. These VPN segments are completely isolated from communicating with each other unless policy explicitly allows communication. These VPNs are carried in a single IPsec tunnel. For example, corporate users could have a full-mesh topology, while PCI or HIPAA requirements could dictate the use of a hub-and-spoke topology for other devices. Figure 2-3 provides a graphical representation of this concept.

On the LAN, or service, side, the data plane supports OSPF, EIGRP, RIPv2,

and BGP for routing protocols. For smaller locations that don’t utilize a routing protocol, VRRP is supported to provide first-hop gateway redundancy.

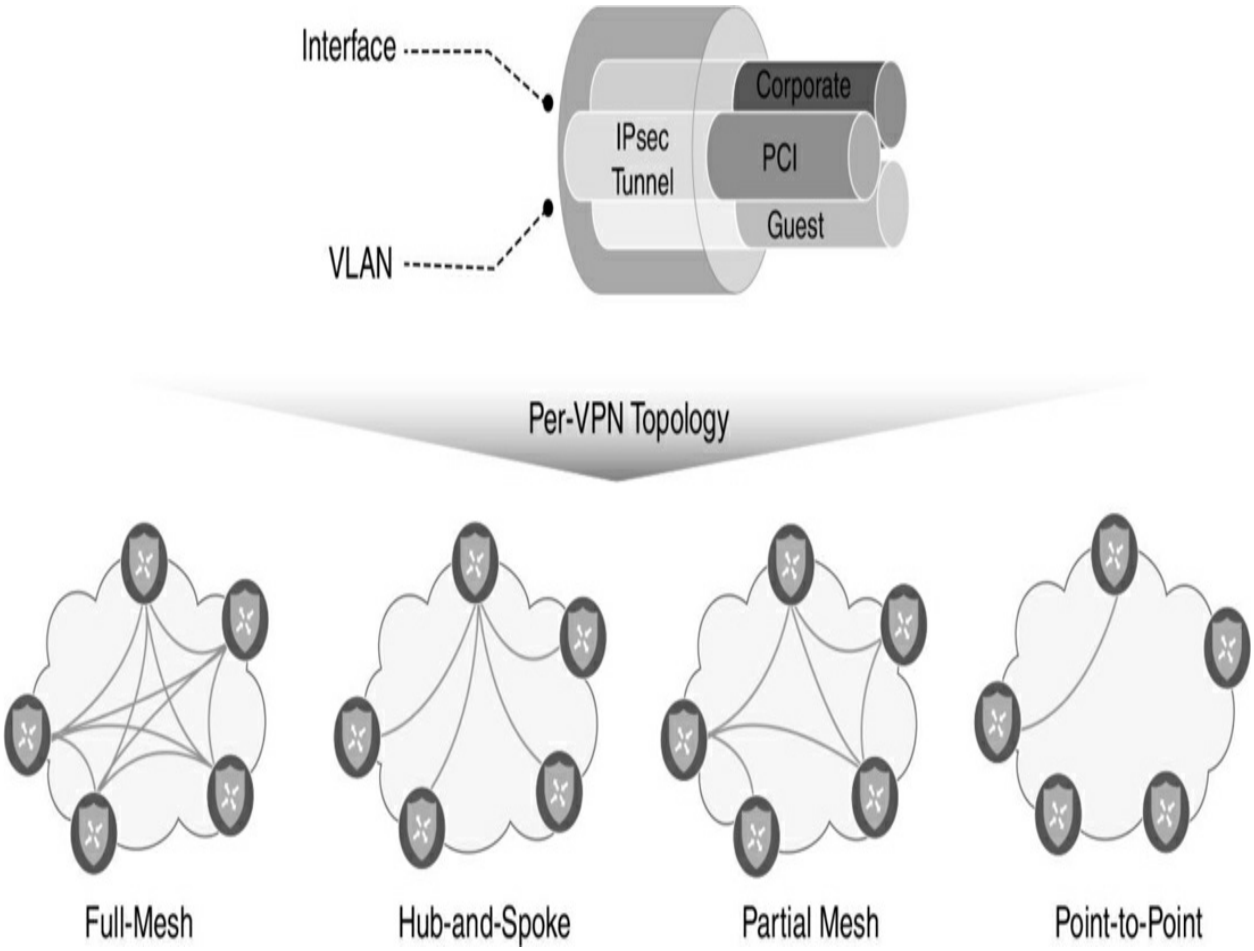


Figure 2-3 *Segmentation and Per-VPN Topologies*

Note

In Cisco Catalyst SD-WAN, Virtual Private Networks (VPN) are synonymous with Virtual Routing and Forwarding (VRF) instances from a generic routing perspective. VRF instances and VPNs provide a method to separate the control and data

Technet24

planes into different logical parts. Segmentation in the data plane is accomplished by building multiple isolated routing table instances and binding specific interfaces to those instances.

WAN Edge routers have built-in security to prevent unauthorized access from the network. The WAN-facing interfaces only allow connections from authenticated SD-WAN fabric components, such as SD-WAN Manager (formerly vManage) and SD-WAN Controllers (formerly vSmarts) and from other WAN Edge devices in the fabric (as learned from SD-WAN Controllers). For the rest of the traffic, the WAN-facing interface firewall on the WAN Edge router, by default, will block everything coming in from the outside that isn't allowed explicitly; this is also called an "implicit ACL." By default, a WAN Edge router only allows inbound DHCP, DNS, ICMP, and HTTP services. Other inbound services that can be enabled are SSH, NETCONF, NTP, OSPF, BGP, SNMP, and STUN.

Note

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

will discover in later chapters.

Bidirectional Forwarding Detection (BFD) is used inside IPsec tunnels between all WAN Edge routers. BFD sends Hello packets to measure link liveness as well as packet loss, jitter, and delay. Each WAN Edge router makes its own determination about how to react to this BFD information. Depending on the policy defined by the management plane, routing across the data plane could be adjusted, such as having applications prefer one transport over the other, depending on the transport performance. BFD operates in echo mode, which means the neighbor doesn't actually participate in the processing of the BFD packet; instead, the BFD packet is simply echoed back to the original sender. This greatly reduces the impact on the CPU, as the neighbor doesn't need to process the packets. However, if the neighbor was involved in the processing of the BFD packets, and the remote neighbor's CPU were busy with some other processing, there could be potential delay in responding to the BFD packet. By eliminating this, you can reduce outage detection time and improve user experience. BFD cannot be turned off, but timers can be tuned in the SD-WAN fabric to identify and illicit a response to potential issues more quickly. Another advantage of using echo mode is that the original packet is echoed back to the original sender, and from this information, the WAN Edge router has a complete round-trip view of the transport.

When the WAN Edge router initially gets connected to the network and has

Technet24

no configuration present, it first tries to reach out to a Plug and Play (PNP) or Zero-Touch Provisioning (ZTP) server. Figure 2-4 provides a high-level overview of the PNP/ZTP process. This process will be discussed further in Chapter 4, "Onboarding and Provisioning," but for now, you just need to know that this is the process in which the router connects to the orchestration plane, learns about all of the various components in the network, and receives its configuration. Once the control plane is established, the last step is to build data plane connections to all other WAN Edge routers. By default, a full-mesh topology will be built, though policy can be built to limit data plane connections and influence the routing topology. It should be noted, as well, that if PNP or ZTP isn't available, there are other options available to manually bootstrap the configuration using the CLI or a USB thumb drive.

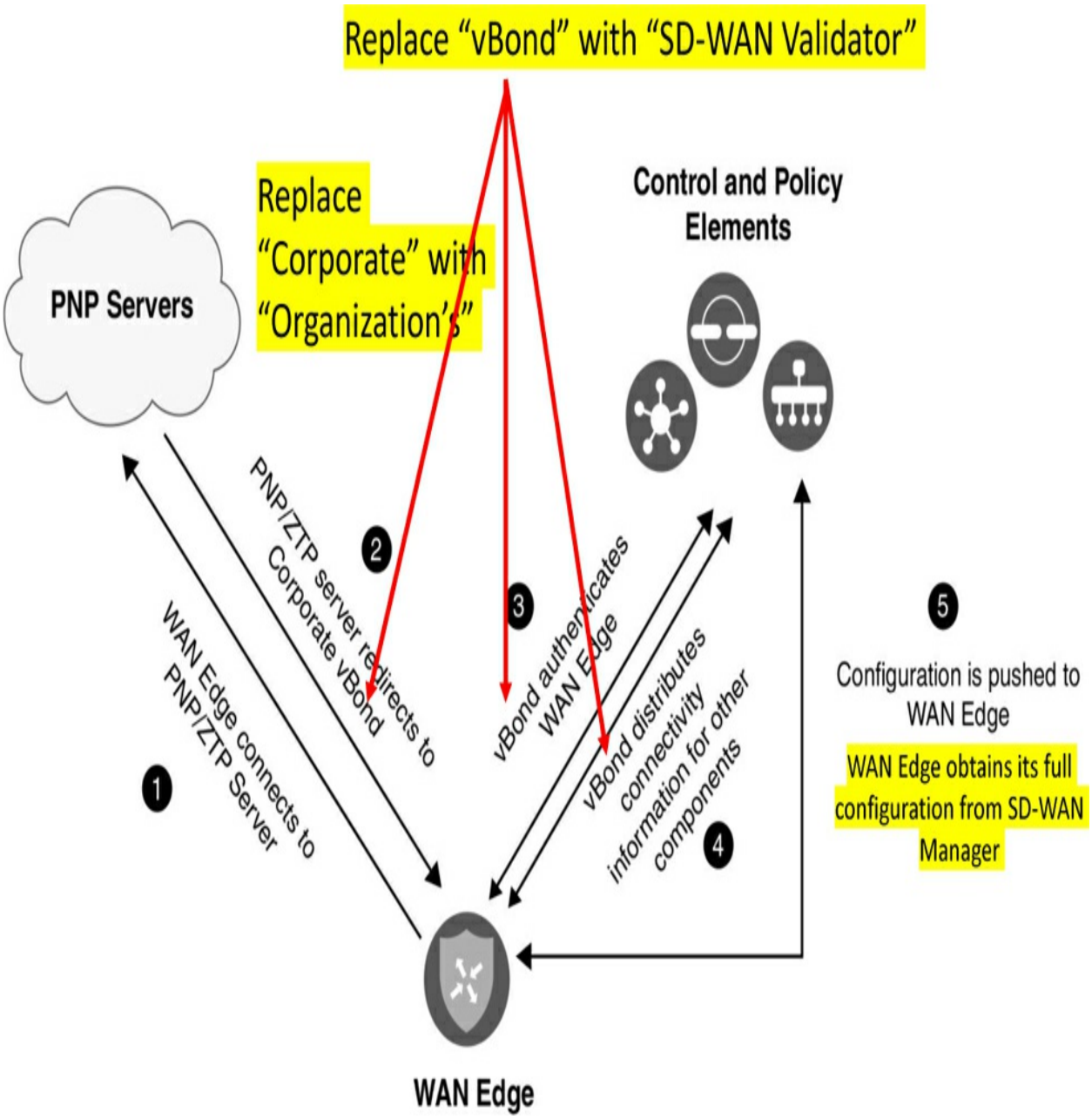


Figure 2-4 *High-Level Overview of the PNP/ZTP Process*

Note

There are two auto-provisioning methods for WAN Edge devices: PNP for Cisco IOS-XE Catalyst SD-WAN routers and ZTP, originally developed for Cisco vEdge routers. While

Technet24

the processes are quite similar, they rely on two different services. Both services are cloud based, but there's also an on-premises deployment option for ZTP that supports both device types. On-premises deployment can help in scenarios where Internet access is not available or local hosting of the PNP functionality is desirable.

SD-WAN Supported Platforms



Cisco offers the wide selection of platforms and appliances so that you can deploy SD-WAN anywhere, as illustrated in Figure 2-5. With Cisco Catalyst SD-WAN, you can create a comprehensive fabric and scale your entire network into hybrid and multicloud environments with ease.

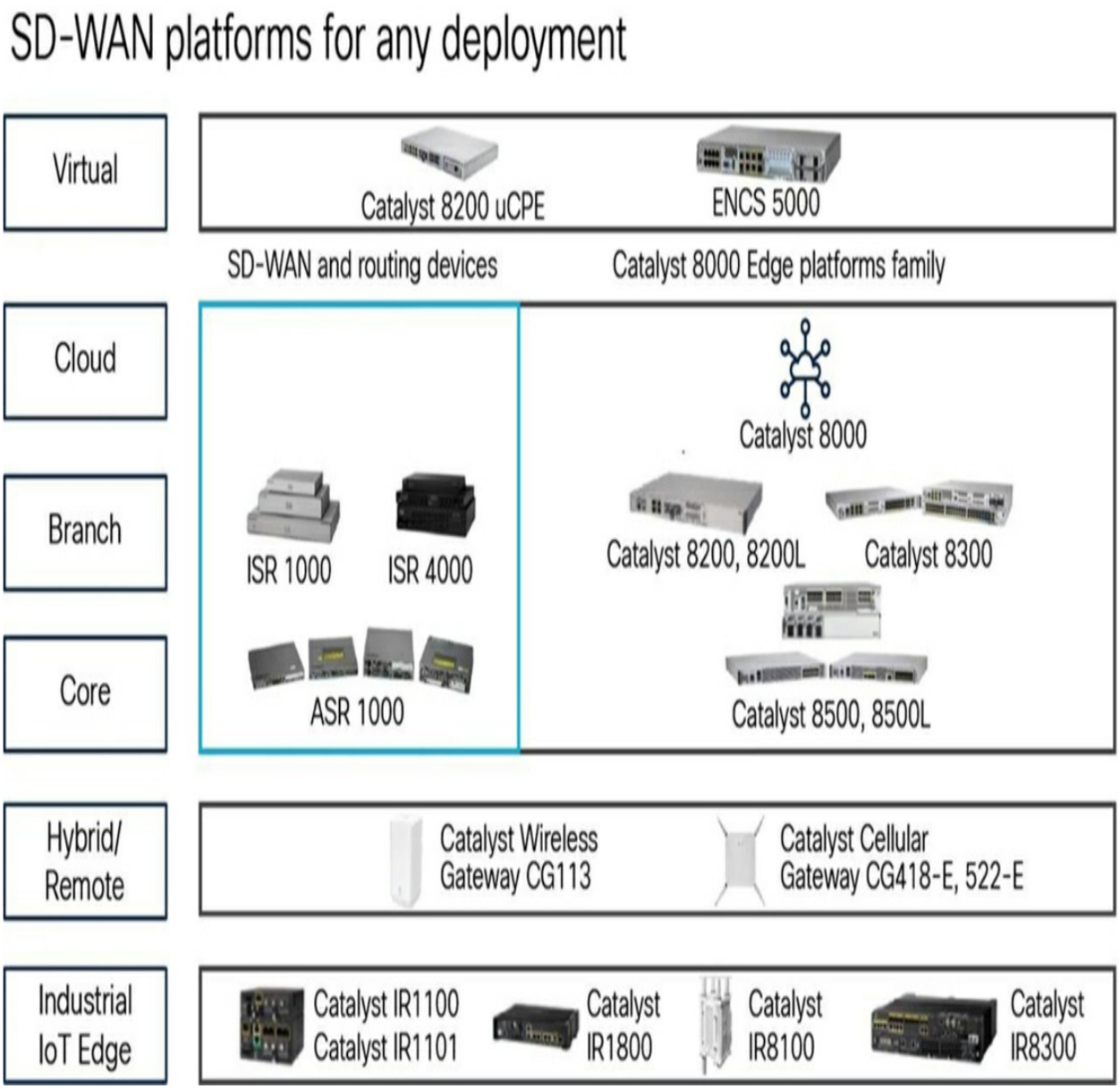


Figure 2-5 *Cisco Catalyst SD-WAN Supported Platforms*

In the ever-evolving landscape of networking, keeping up with product offerings can be challenging. At this writing, the leading Cisco Catalyst SD-WAN hardware platforms include Cisco Catalyst 8500, 8300, and 8200 Series Edge platforms, along with Cisco 1100 Series Integrated Services Routers (ISRs). Cisco Catalyst SD-WAN can also be deployed on SD-Branch

Technet24

solutions such as the Catalyst 8200 Series Edge uCPE and Cisco 5000 Enterprise Network Compute System (ENCS) using Network Functions Virtualization (NFV).

While Cisco Catalyst SD-WAN remains compatible with earlier hardware generations, like Cisco 4000 Series ISRs, Cisco Advanced Services Routers (ASRs), and original vEdge routers, it is important to note that these platforms are at different stages of their end-of-life lifecycle. Consequently, they are not recommended for new deployments.

A noteworthy feature of Cisco Catalyst SD-WAN is its seamless integration with public cloud environments, made possible through the Cisco Catalyst 8000V Edge Software and the legacy Cloud Services Router 1000V Series virtual platforms. Deployments in Amazon Web Services, Google Cloud, and Microsoft Azure are supported, offering flexibility and scalability. Virtual platforms can also be deployed in private clouds, running on either VMware ESXi or KVM hypervisors.

When deciding on WAN Edge platform, it’s crucial to assess your specific requirements, including deployment type (physical or virtual), throughput needs, data plane tunnel scalability (such as the number of branches the router will communicate with), and the required interface types.

Note

Information about the Cisco Catalyst SD-WAN platform is

accurate as of this writing. For the most current details, please consult the Cisco Catalyst SD-WAN home page or refer to the “SD-WAN Platforms” section in the Cisco Catalyst SD-WAN Solution Overview document on the Cisco documentation site.

Note

While Cisco Catalyst SD-WAN continues to support the original Cisco Viptela vEdge hardware models, including vEdge 100, vEdge 1000, vEdge 2000, and vEdge 5000, it is imperative to acknowledge the announced end of life for these devices. (For specific milestones and dates, please consult the product end-of-sale and end-of-life announcements.) Consequently, software development for this platform is concluding, with Version 20.6 marking the final release of Cisco Catalyst SD-WAN that includes support for Cisco vEdge 100 and vEdge 1000 devices. Version 20.9 is the last release for Cisco vEdge 2000 and vEdge 5000 routers.

Given these changes, this book focuses on the Cisco IOS XE SD-WAN platforms. All functionalities and features are described as they are implemented on Cisco IOS XE devices. Throughout the book, examples exclusively use Cisco IOS XE commands and configuration syntax, unless explicitly stated otherwise.

Technet24

Some of the most important features supported on Cisco Catalyst SD-WAN routers are for advanced security use cases. While accessing the Internet directly via a local Internet circuit might pose security risks at the branch, overlaying security on top of Cisco Catalyst SD-WAN enables safe implementation of new use cases such as Direct Internet Access (DIA) and Direct Cloud Access (DCA).

Direct Internet Access allows certain Internet-bound traffic (for example, Facebook traffic, YouTube traffic) to be forwarded from the branch directly to the Internet instead of being backhauled to data center via SD-WAN fabric. Direct Cloud Access enables cloud traffic (for example, Office365 traffic, Salesforce traffic, Box traffic, Google traffic) to be sent from the branch directly to the Internet or, optionally, backhauled to data centers based on path performance. Figure 2-6 illustrates these concepts.

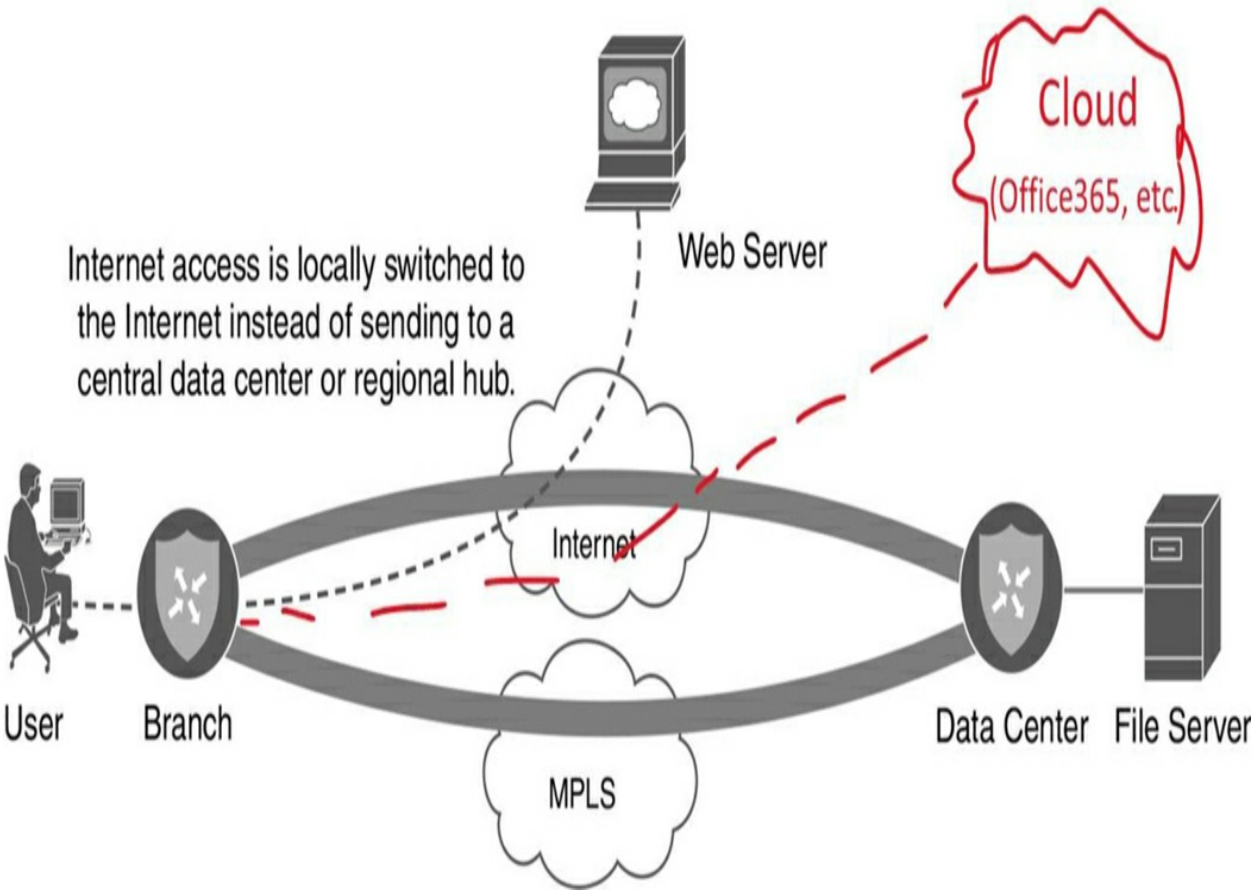


Figure 2-6 *Direct Internet Access/Direct Cloud Access Overview*

Security use cases are discussed in more detail in Chapter 11, "Cisco Catalyst SD-WAN Security," but here is a list of some of the currently supported security features:

• DNS security (Cisco Umbrella)

• Secure Internet Gateways (SIGs) and Cisco Secure Access integration

• Cisco Enterprise Firewall with Application Awareness

• Intrusion Prevention Systems (IPSs)

Technet24

• URL Filtering

• Cisco Advanced Malware Protection

• SSL/TLS Proxy for Decryption of TLS Traffic

Traditionally, security requirements dictated centralization of the Internet access where all Internet traffic is backhauled to data centers, colocation, or regional sites. It was cost-effective to implement security at a central location rather than deal with the management and costs of disparate security components across many sites. With Cisco Catalyst SD-WAN security, businesses can now decentralize security functions, moving them to the branch level. This movement allows organization to offload Internet access at remote sites. Here are some other areas where a business might see benefits from DIA:

• Reduced bandwidth requirements and latency on costly WAN circuits

• Guest access

• Improved user experience to cloud SaaS and IaaS applications

Chapter 8, "Centralized Data Policies," discusses DIA in more detail.

When a WAN Edge router joins the fabric, it attempts to build control connections to SD-WAN Control Components across each transport deployed at that site. By default, if a transport doesn't have control

connectivity to any of the SD-WAN Control Components, then it won't build a data plane connection across that transport either. This may be the case with cloud deployments where the controllers are in a public cloud and MPLS transport has no connectivity to the Internet.

Note

There are a few options to still activate the data plane for a transport with no control connectivity. One option is to disable control connections on that transport via the **max-control-connections 0** command. Be aware that when control connections aren't established on an interface, there will be no control plane monitoring over that transport. You still have monitoring from a data plane perspective, however.



Management Plane

As mentioned previously, network devices of the past were managed individually via the CLI. Cisco Catalyst SD-WAN, however, introduces SD-WAN Manager (formerly vManage), which is a network management system (NMS) that provides a single pane of glass to manage Catalyst SD-WAN. SD-WAN Manager can be used for device onboarding, provisioning, policy creation, software management, troubleshooting, and monitoring.

Technet24

While SD-WAN Manager offers a rich feature set, if the preference is to interface with it programmatically, SD-WAN Manager also supports communication via REST APIs. In fact, the SD-WAN Manager GUI is fully API driven, meaning that actions performed in it are executed using REST API calls. With full access to SD-WAN APIs, users can automate tasks, build scripts, and interface with SD-WAN Manager programmatically.

As you can see in Figure 2-7, SD-WAN Manager provides an intuitive and easy-to-consume dashboard. When you first log in to SD-WAN Manager, you are presented with an overview of the current state of the network.

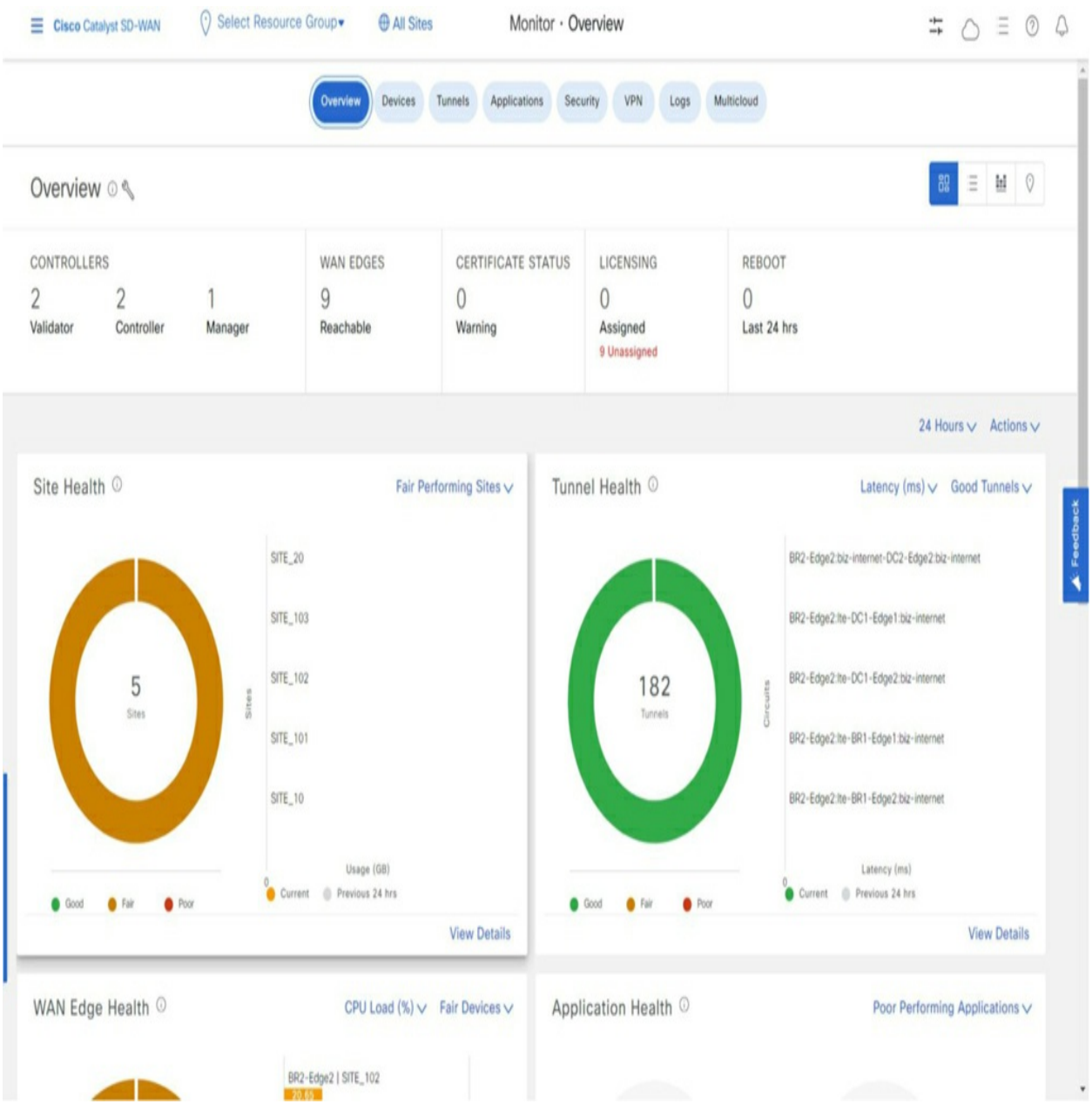


Figure 2-7 *Cisco SD-WAN Manager*

vManage deployment options range from standalone nodes to three- or six-node clustered setups, offering enhanced scale and redundancy. A single SD-WAN Manager can potentially handle up to 1000 to 1500 devices, and a six-node SD-WAN Manager cluster may support more than 10,000 devices. It’s important to note that these numbers may vary based on a number of factors,

Technet24

such as SD-WAN Manager resources (instances/CPU/RAM/storage), the statistics load, and the version of SD-WAN software in use. (Numbers mentioned in this chapter are specific to Version 20.12.) For more accurate specifications, please consult the “Recommended Computing Resources for Cisco Catalyst SD-WAN Control Components” document for the SD-WAN software version you are using or are planning to use, available on the Cisco website.

An SD-WAN Manager cluster is designed to tolerate the failure of a single server, but for high availability, a standby cluster should be implemented to handle a complete cluster failure. Typically, it is deployed in a geographically redundant location, such as a secondary data center in another region.SD-WAN Manager can use multiple authentication sources, including RADIUS, TACACS, and SAML 2.0, for external user connectivity. By default, SD-WAN Manager is deployed in a single-tenant mode; however, if the requirements call for support of a service provider model, multi-tenancy may be used.

All configuration for the SD-WAN fabric should be performed via SD-WAN Manager in order to maintain consistency and scalability. As discussed further in Chapter 4, you can build device configurations in SD-WAN Manager via configuration groups, feature templates, or CLI templates. You can also configure policies to control things such as network topology, routing, QoS, and security in SD-WAN Manager. SD-WAN Manager is also where you perform troubleshooting and monitoring of the network. Network

administrators can simulate traffic flows to show data paths, troubleshoot WAN impairment, analyze traffic flows in the network with Network-Wide Path Insights (NWPI), and access real-time operational information (such as routing tables) for all network devices. This greatly simplifies operations as there is no longer a need to log in to each WAN Edge router individually. Instead, troubleshooting can be accomplished via a single dashboard.

Each WAN Edge router forms a single management plane connection to SD-WAN Manager. If a device has multiple transports available, only one will be used for management plane connectivity to SD-WAN Manager. If a cluster is in place, the control connection will be load balanced across cluster nodes. If a transport hosting the management plane connection experiences an outage, the WAN Edge router will briefly lose connectivity to SD-WAN Manager, and any changes made will be pushed when the device reconnects.

The last component in the management plane is SD-WAN Analytics (formerly vAnalytics). As shown in Figure 2-8, SD-WAN Analytics gives the network administrator predictive analytics to provide actionable insight into the WAN. With SD-WAN Analytics, the business can perform trending and capacity planning of circuits, and it can review how application performance is trending globally. With capacity planning, you can see how new applications may interact on your WAN before actually deploying them, allowing your business to right-size connectivity. SD-WAN Analytics ingests data from the network and uses machine learning to predict capacity trends. SD-WAN Analytics is cloud based, it requires additional licensing, and it is

Technet24

not enabled by default.

Note

It is important to note that SD-WAN Manager should be used for a real-time, raw data view of the network, while SD-WAN Analytics should be used as a tool to review the historical performance of the network and get forward-looking insight into network adjustments.

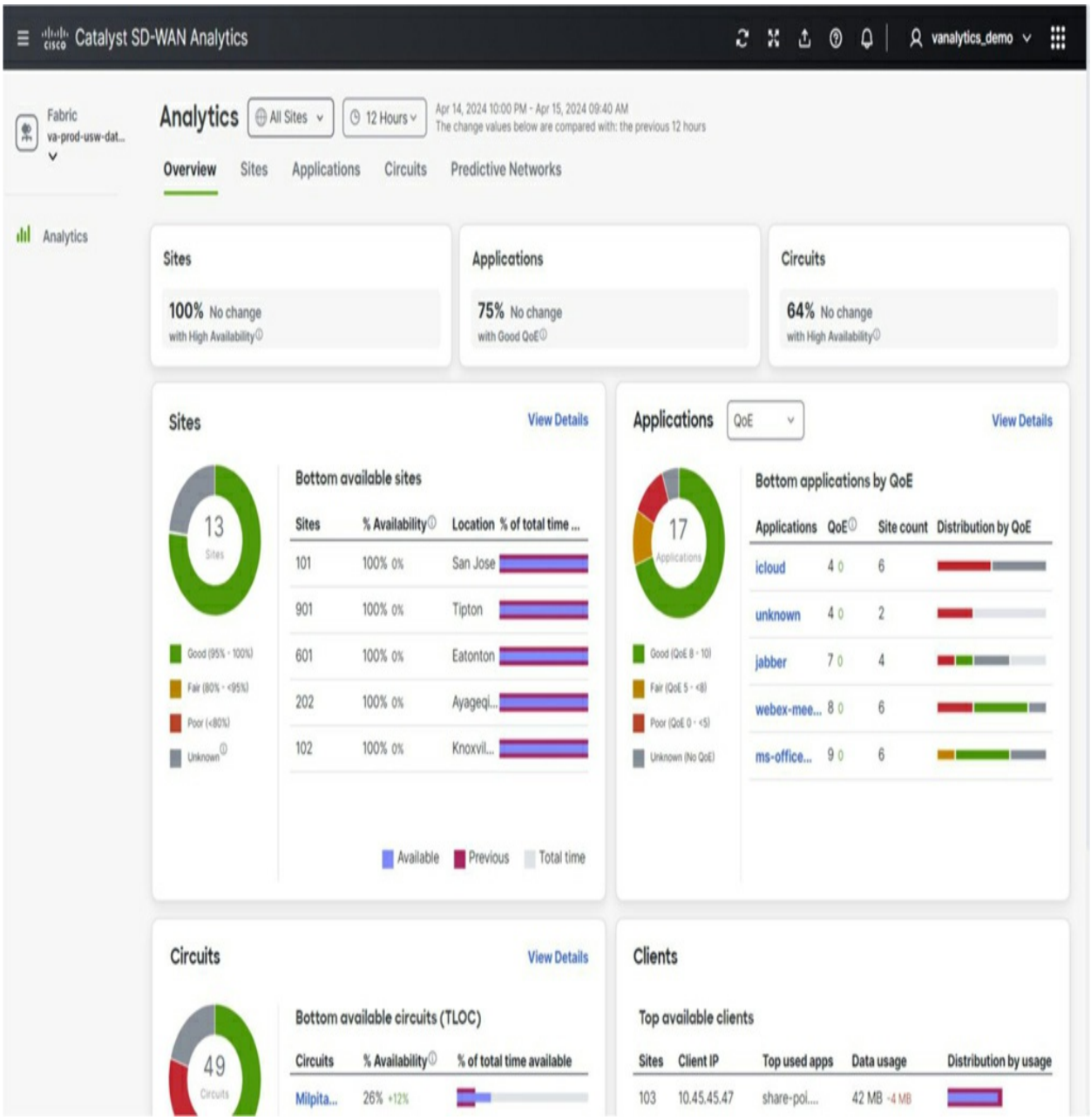


Figure 2-8 *Cisco SD-WAN Analytics*

Key Topic

Control Plane

Previously, you learned how the control plane has traditionally been

Technet24

separated from the data plane. SD-WAN Controller (vSmart) is the component that provides control plane functionality and is the brain of the SD-WAN fabric. It is highly scalable and can handle up to 5000 control connections per instance, with up to 12 SD-WAN Controllers in a single production deployment (as of SD-WAN Version 20.12). With these numbers, a deployment can support very large SD-WAN networks.

SD-WAN Controller is responsible for the implementation of control plane policies, centralized data polices, service chaining, and VPN topologies. It also handles key management which is an important part in the security and encryption of the fabric.

Separating the control plane from the data and management planes allows a solution to achieve greater scale while simplifying network operations. With Cisco Catalyst SD-WAN, all SD-WAN Controllers learn all the routing information. Then they calculate the routing table and distribute it to the WAN Edge routers, taking into consideration applicable centralized control policies.

A WAN Edge router can connect to multiple SD-WAN Controllers at a time but needs connectivity to only one to get its routing and policy information.

The protocol that SD-WAN Controllers use to communicate all this information is called *Overlay Management Protocol (OMP)*. Although OMP handles routing, it would be a disservice to consider it simply a routing

protocol. OMP is used to manage and control the overlay beyond just routing (key management, configuration updates, and so on). As illustrated in Figure 2-9, OMP runs between SD-WAN Controller and WAN Edge routers inside a secured tunnel. When a policy is built via the management plane, it is distributed to SD-WAN Controller via NETCONF, and SD-WAN Controller then distributes this policy via an OMP update to WAN Edge routers.

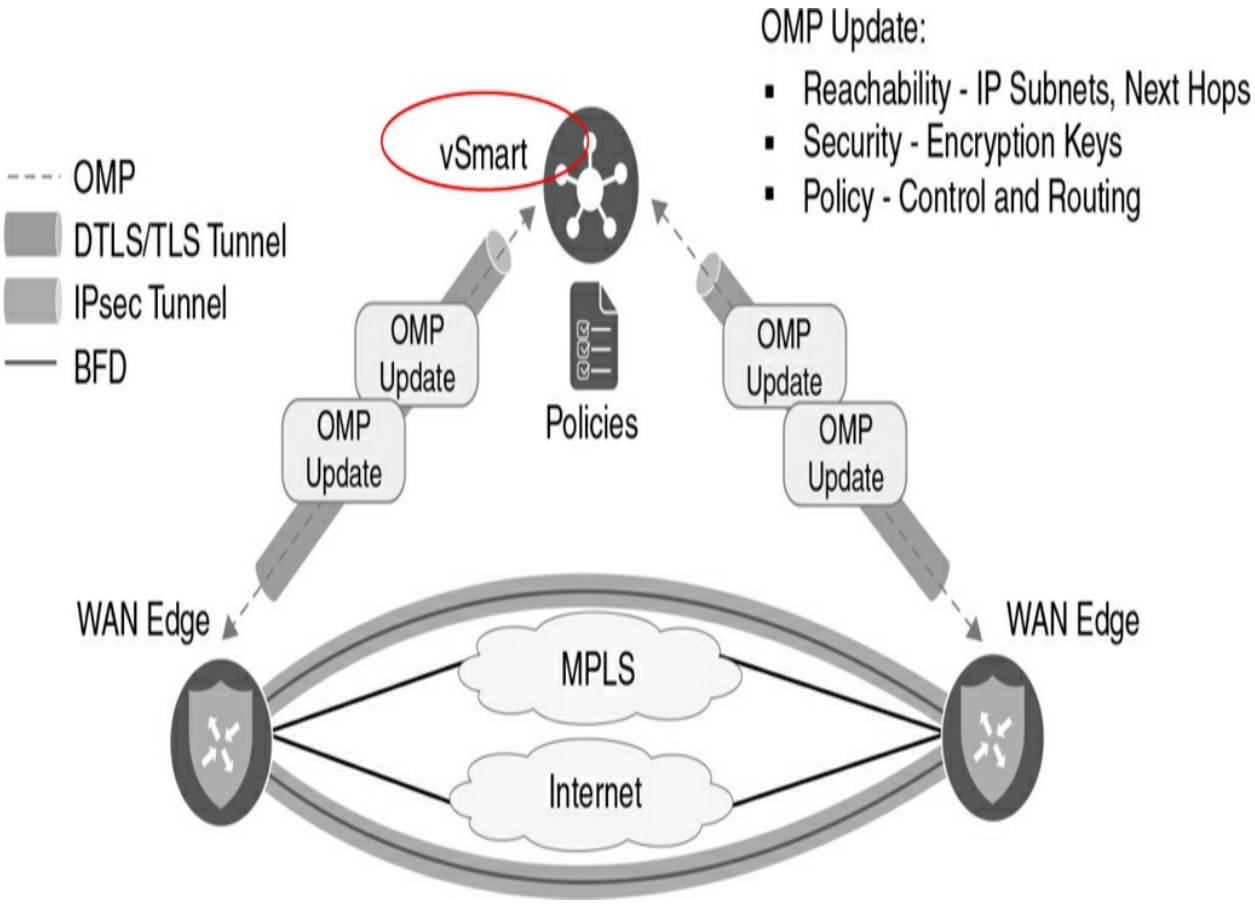


Figure 2-9 *Cisco Control Plane and Data Plane Overview*

SD-WAN Controller operates similarly to a BGP route reflector in iBGP. It receives routing information from each WAN Edge router and can apply policies before advertising this information back out to other WAN Edge

Technet24

routers. One example of these policies is the creation of distinct per-VPN topologies. To achieve it, the control policy is defined in SD-WAN Manager, which then distributes the policy through the management plane. Then, SD-WAN Controller applies the policy to the fabric. In this example, topology modification is achieved by manipulating what routes get distributed and how the data plane is built between WAN Edge routers.

Note

Note that SD-WAN Controllers are only involved in control plane communication. SD-WAN Controllers help WAN Edge routers build the data plane but are never a part of the data plane and never forward data packets.

The control plane also plays an important role in the encryption of the fabric. In legacy WAN technologies, securing the network required a considerable amount of processing power, as each device would compute its own encryption keys per peer and distribute those keys to peers by using a protocol such as ISAKMP/IKE. For more efficient scaling in Cisco SD-WAN networks, key exchange and distribution have been moved to the SD-WAN Controller, and no IKE is implemented since identity has already been established between the WAN Edge routers and SD-WAN Control Components. Each WAN Edge router computes its own set of keys per transport and sends them to SD-WAN Controllers. SD-WAN Controllers then distribute them to each WAN Edge router, according to the defined

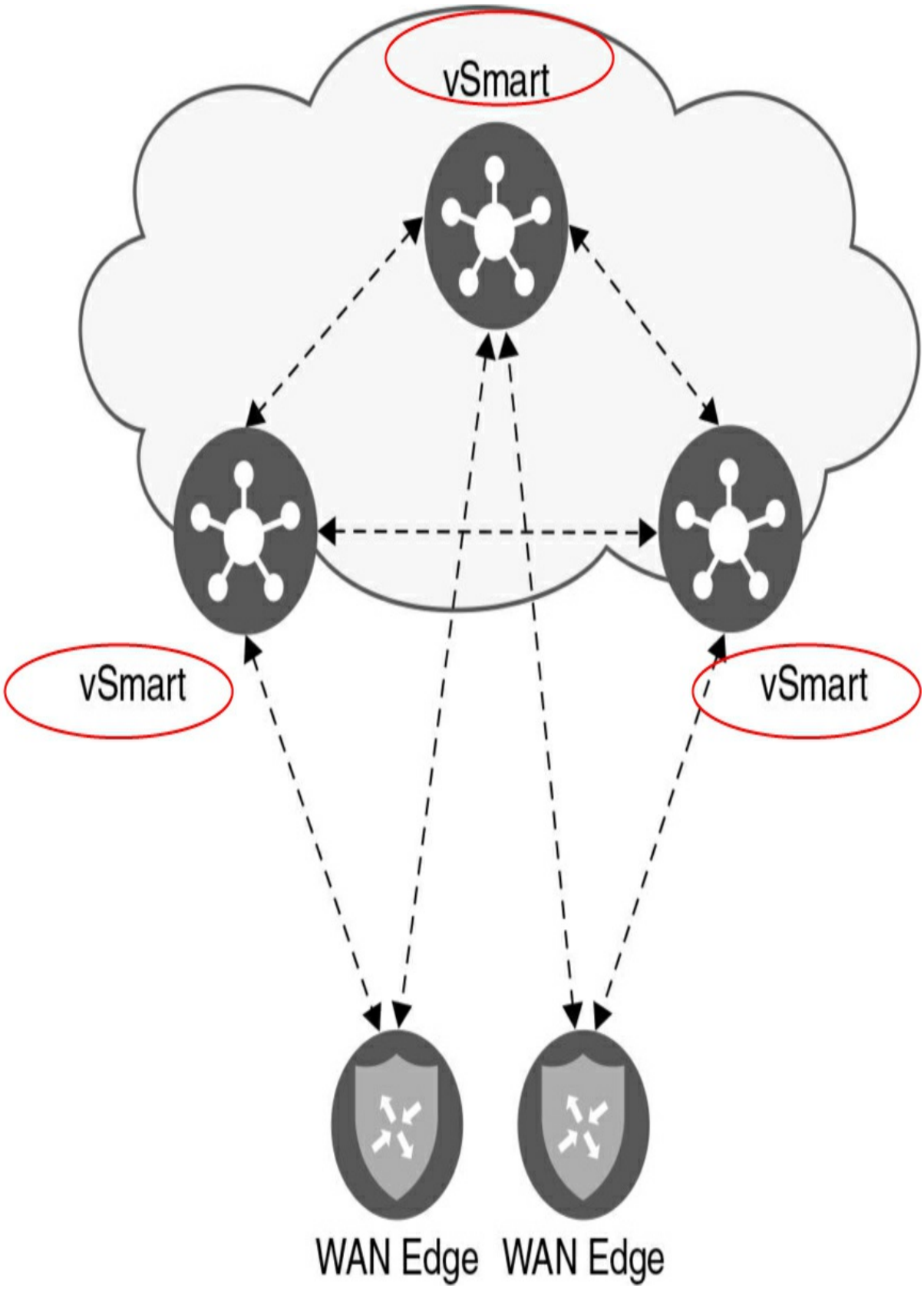
policy. This process repeats when IPsec security associations (SAs) expire and new keys are generated. By moving the key exchange to a centralized location, you achieve greater scale as each WAN Edge router doesn't need to handle key negotiation or distribution. (Refer to Figure 2-9 for an overview of how the control and data planes are built.) Chapter 3, "Control Plane and Data Plane Operations," covers this in more detail.

If control connectivity has been established but has been lost due to an outage, data plane connectivity continues to work. By default, WAN Edge routers continue forwarding data plane traffic in the absence of control plane connectivity for up to 12 hours, utilizing the last-known state of the routing table, although this is configurable, depending on your requirements. When control plane connectivity is reestablished, WAN Edge routers are updated with any policy changes that were made during the outage. When the control connection is restored, the routing table is refreshed, and any stale routes are flushed.

For redundancy, it is the best practice to deploy at least two geographically dispersed SD-WAN Controllers. They should have identical policy configuration to ensure network stability. If these configurations differ, there's a risk of suboptimal routing and potential blackholing of traffic. SD-WAN Controllers maintain a full mesh of OMP sessions among themselves and exchange control and routing information, although each operates autonomously (that is, there is no database synchronized between the two).

Technet24

Figure 2-10 shows how OMP is established between multiple SD-WAN Controllers and WAN Edge routers. SD-WAN Controllers form a full mesh among themselves, which ensures that they stay synchronized. WAN Edge routers establish one OMP session to each of two (by default) SD-WAN Controllers, but they do not create OMP sessions with each other. When there are more than two SD-WAN Controllers in the network, their selection is based on an algorithm to ensure that load balancing occurs. In the event of the total failure of SD-WAN Controller, the sessions are redistributed between the remaining SD-WAN Controllers to maintain network continuity and stability.



Technet24

Figure 2-10 *OMP Session Establishment*

By default, each WAN Edge router establishes multiple secure control connections, one over each available transport, to each selected SD-WAN Controller. However, only one OMP session is established between the WAN Edge device and each SD-WAN Controller, using one of those control connections as a transport.



Orchestration Plane

The final, and probably the most important, component in Cisco Catalyst SD-WAN is SD-WAN Validator (formerly vBond). This component is very important because it provides initial authentication for participation in the fabric and acts as the glue that discovers and brings together all other components. Multiple SD-WAN Validator instances can be deployed to achieve high availability. Since a WAN Edge router can point to only a single SD-WAN Validator, it is recommended to configure WAN Edge routers to use a DNS name that has a single A record pointing to the IP addresses of all SD-WAN Validators. When the WAN Edge router tries to resolve the DNS record for the SD-WAN Validator, the DNS server provides multiple IP addresses for the hostname, and the WAN Edge router tries to connect to each of them sequentially until a successful control connection is made.

When a WAN Edge router first joins the overlay, the only thing it knows

about the SD-WAN network is IP address or DNS name of the SD-WAN Validator. It receives this information via one of these methods:

• PNP/ZTP

• Bootstrap configuration

• Manual configuration

The WAN Edge router attempts to build a temporary connection to the SD-WAN Validator over each transport. Once the control plane connectivity is up to SD-WAN Controller and SD-WAN Manager, the connection to the SD-WAN Validator is torn down. When the WAN Edge router connects to the SD-WAN Validator, they both go through an authentication process in which each component authenticates the other and, if successful, a Datagram Transport Layer Security (DTLS) tunnel is established. The SD-WAN Validator then distributes the connectivity information for the SD-WAN Controller and SD-WAN Manager to the WAN Edge router. You can see why the SD-WAN Validator is referred to as the glue of the network: It tells all the components about each other. (This process is discussed in more detail in Chapters 3 and 4.)

One remaining functionality that the SD-WAN Validator provides is NAT traversal. By default, the SD-WAN Validator operates as a STUN server (RFC 5389). A WAN Edge router operates as a STUN client. What this means is that the SD-WAN Validator can detect when WAN Edge routers are

Technet24

behind a NAT device such as a firewall. When a WAN Edge router goes to establish its DTLS tunnel, the interface IP address it knows about is written into the outer IP header and noted within a payload of the message. When SD-WAN Validator receives this information, it compares the two values. If they are different, it can be inferred that NAT is in the transit path of the WAN Edge router (since the outer IP header was changed to a NAT IP address and no longer matches the IP address noted in the payload of the packet). The SD-WAN Validator communicates this back to the WAN Edge router, and the WAN Edge router can communicate this information to the rest of the overlay components—ultimately allowing data plane connectivity to be established through a NAT device. There are, however, some scenarios where this won't work, such as with symmetric NAT (as discussed in more detail in Chapters 3 and 4). Figure 2-11 explains how STUN is used to detect when a WAN Edge router or another SD-WAN Control Component is subject to NAT.

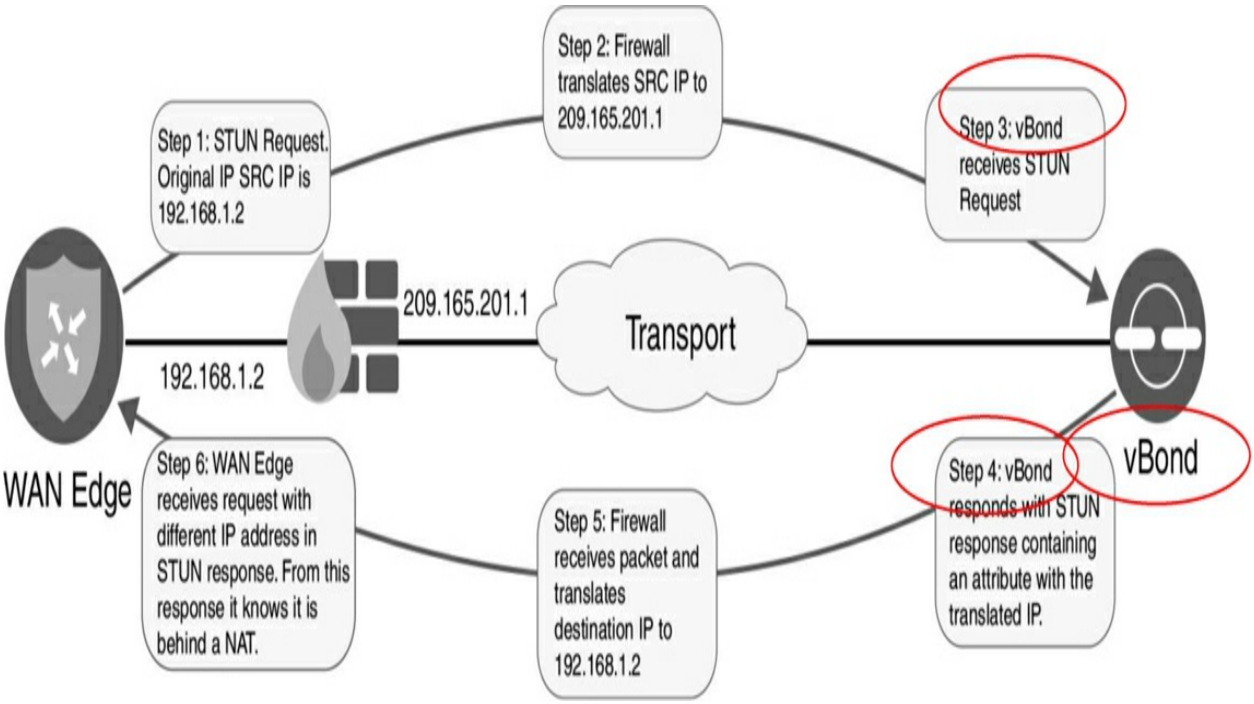


Figure 2-11 *STUN NAT Detection Method*

When deploying an SD-WAN Validator, one special consideration is that it must be reachable directly from all transports with enabled control connectivity. This implies that SD-WAN Validator needs a public IP address (or a private IP address with 1:1 static NAT) when Internet transports are used in SD-WAN fabric.

Other SD-WAN Control Components (including SD-WAN Managers and SD-WAN Controllers) can use private IP addresses as long as they have connectivity to SD-WAN Validator since they use the same NAT discovery method (STUN) as the WAN Edge routers.

Multi-tenancy Options

Cisco Catalyst SD-WAN supports multiple modes of segmentation in the

Technet24

control, data, management, and orchestration planes, as shown in Figure 2-12. One mode is dedicated tenancy. In this mode, each tenant has dedicated components, and the data plane is segmented as well. The second option is VPN tenancy. This mode segments only the data plane of the VPN topology and allows you to define read-only users who can view and monitor their VPN within SD-WAN Manager. VPN tenancy still shares the same SD-WAN components, however. The third option is multi-tenancy. With Cisco Catalyst SD-WAN multi-tenancy, a service provider can manage multiple customers, called tenants, from SD-WAN Manager. The tenants share the same set of underlying SD-WAN Control Components: SD-WAN Manager, SD-WAN Validators, and SD-WAN Controllers. The tenant data is logically isolated on these shared SD-WAN Control Components. WAN Edge devices are typically tenant specific (that is, not shared), but service providers managing a multi-tenant SD-WAN deployment may deploy a multi-tenant WAN Edge devices to serve as a shared gateway for traffic belonging to multiple tenants.

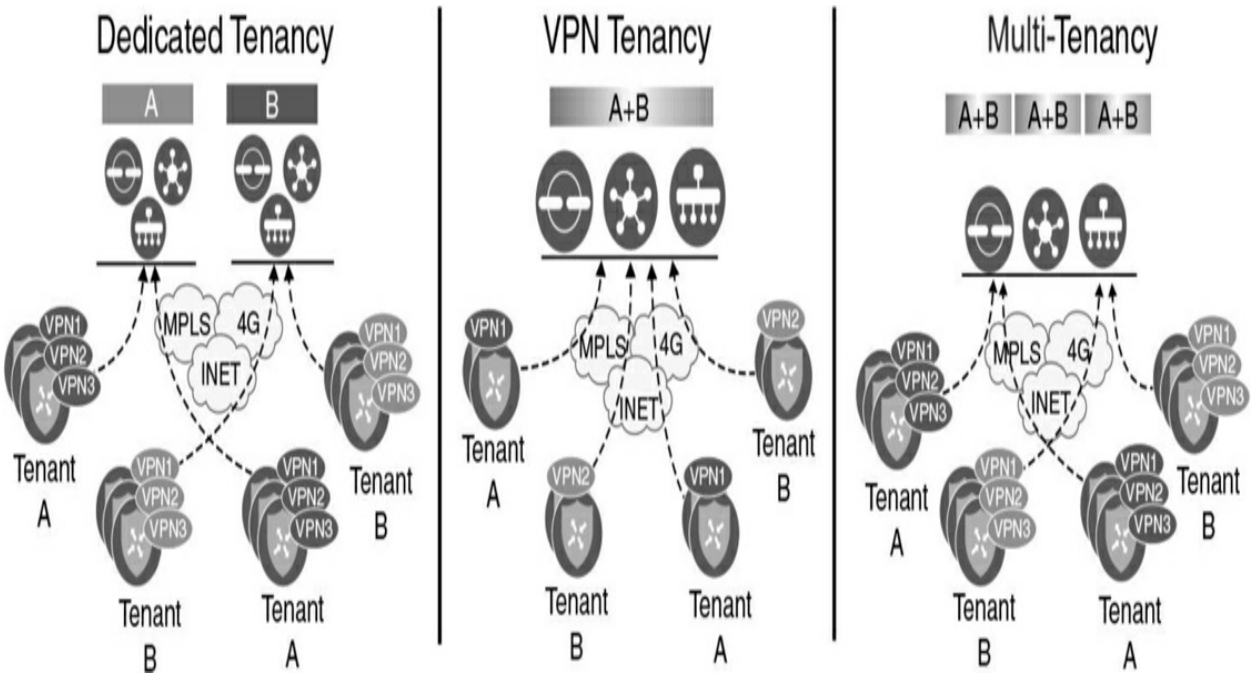


Figure 2-12 *Cisco Catalyst SD-WAN Multi-Tenancy Options*

Deployment Options

Cisco Catalyst SD-WAN supports multiple deployment options for SD-WAN Control Components:

• **Cisco cloud hosted:** This is the recommended and the most common deployment option, where Cisco builds, operates, and monitors all the SD-WAN Control Components introduced in this chapter. This greatly simplifies the deployment, allowing the network administrator to focus on the configuration and policy administration of the Cisco Catalyst SD-WAN fabric. Customers can manage their deployments through the Cisco Catalyst SD-WAN portal (https://ssp.sdwan.cisco.com), a cloud-infrastructure automation tool tailored for Cisco Catalyst SD-WAN. Using this portal, they can submit fabric provision requests, indicating the preferred cloud provider

Technet24

(AWS or Azure) and regions for hosting SD-WAN Control Components and data. The SD-WAN portal also offers additional functions, such as controller monitoring and fabric maintenance.

• **Managed service provider (MSP) or partner hosted:** SD-WAN Control Components are hosted in a private or public (AWS or Azure) cloud of an MSP or partner. The provider is responsible for provisioning of the SD-WAN Control Components and backups/disaster recovery.

• **On premises:** This option is suitable when business requirements dictate hosting SD-WAN Control Components in a traditional data center. With this approach, customers assume full responsibility for deploying, managing, and monitoring the SD-WAN Control Components. While it offers full control, customers must also manage the operations and maintenance of servers hosting these components. Another challenge is resource scaling, which is not as easy as with cloud-hosted solutions. On-premises deployment is particularly important for companies that are subject to strict regulations, such as those in the government, financial, healthcare, and utilities sectors.

• **Customer cloud hosted:** In this mix of previous options, SD-WAN Control Components are deployed in a public cloud (AWS or Azure), but customers are fully responsible for their deployment, management, and monitoring. Compared to on-premises hosting, customer cloud deployments have a low initial setup cost, as there is no need to purchase additional data center infrastructure. This option brings in traditional cloud benefits: ease of

provisioning, stability, security, and scaling.

Summary

This chapter introduces the components that make up Cisco Catalyst SD-WAN. It discusses the data plane, wherein user traffic is routed and forwarded across the WAN. The data plane is similar to routers that would be deployed in a traditional WAN; in Cisco Catalyst SD-WAN, they are referred to as WAN Edge routers.

This chapter also introduces SD-WAN Manager, which is part of the management plane, where all Day 0, Day 1, and Day N functions are performed, including WAN Edge configuration, routing and control policies, troubleshooting, and monitoring.

You have seen that SD-WAN Controller, the brain of the Cisco Catalyst SD-WAN fabric, is responsible for calculating and deploying all control and data policies as well as handling the distribution of encryption keys for data plane connectivity.

You have also seen that SD-WAN Validator makes up the orchestration plane and is responsible for authenticating components on the fabric in addition to distributing control and management plane information to the WAN Edge routers. SD-WAN Validator is the component that aids in discovery of the fabric for all other components (such as when devices are behind NAT).

Technet24

Finally, this chapter discusses deployment options. The most common deployment method is Cisco cloud hosted, but there are three other options to consider: on-premises, public cloud, and customer cloud hosted. By supporting all four deployment options, Cisco Catalyst SD-WAN can support all business requirements.

Review All Key Topics

Review the most important topics in the chapter, noted with the Key Topic icon in the outer margin of the page. Table 2-2 lists these key topics and the page number on which each is found.



Table 2-2 *Key Topics*

Key Topic Element	Description	Page
Figure 2-1	Cisco Catalyst SD-WAN distributed architecture	
Section	Data plane	
Section	Cisco Catalyst SD-WAN supported platforms	
Section	Management plane	
Section	Control plane	
Section	Orchestration plane	

Key Terms

Define the following key terms from this chapter and check your answers in the glossary:

control plane

data plane

Technet24

management plane

orchestration plane

Overlay Management Protocol (OMP)

SD-WAN Controller

SD-WAN Manager

SD-WAN Validator

WAN Edge

Chapter Review Questions

1. What are the three Cisco Catalyst SD-WAN Control Components? (Choose three.)

a. SD-WAN Controller

b. SD-WAN Validator

c. WAN Edge

d. SD-WAN Manager

e. SD-WAN Orchestrator

2. How does the Cisco Catalyst SD-WAN architecture differ from traditional WAN technologies? (Choose three.)

a. Single pane of glass

b. Increased scale with centralized control plane

c. Reduced uptime in branch locations

d. Topology dependence

e. Distributed architecture

3. What are the three functions of SD-WAN Manager in Cisco Catalyst SD-WAN?

a. Troubleshooting

b. Configuration

c. Redistribution

d. Loop prevention

e. Monitoring

4. True or false: WAN Edge routers provide data plane encryption via IPsec.

Technet24

a. True

b. False

5. What traditional networking concept does SD-WAN Controller closely relate to?

a. OSPF designated router

b. DHCP helper

c. BGP route reflector

d. PIM designated router

6. What functions does SD-WAN Validator provide in the SD-WAN environment? (Choose two.)

a. Authentication and authorization of the SD-WAN components

b. NAT detection and traversal

c. Pushing configuration to WAN Edge routers

d. Software upgrades

7. True or false: Cisco Catalyst SD-WAN supports multi-tenancy.

a. True

b. False

8. Which routing protocols are not supported on the service side of Cisco Catalyst SD-WAN? (Choose two.)

a. EIGRP

b. OSPF

c. RIPv1

d. OMP

e. BGP

9. What attributes are measured with BFD? (Choose three.)

a. Delay

b. Loss

c. Jitter

d. Out-of-order packets

10. True or false: Cisco Catalyst SD-WAN is able to provide segmentation

Technet24

and different topologies per VPN.

a. True

b. False

11. Which is not a valid option for the deployment of SD-WAN Control Components?

a. Cisco cloud hosted

b. Customer on premises

c. Partner-cloud hosted

d. Cisco on premises

References

RFC 4023, “Encapsulating MPLS in IP or Generic Routing Encapsulation (GRE),” https://tools.ietf.org/html/rfc4023, March 2005.

RFC 5389, “Session Traversal Utilities for NAT (STUN),” https://tools.ietf.org/html/rfc5389, October 2008.

“Cisco Catalyst SD-WAN,” https://www.cisco.com/site/us/en/solutions/networking/sdwan/index.html.

“SD-WAN Platforms,” https://www.cisco.com/c/en/us/solutions/collateral/enterprise-networks/sd-wan/nb-06-sd-wan-sol-overview-cte-en.html#SDWANplatforms.

Technet24

Chapter 3. Cisco SD-WAN Control Plane and Data Plane [This content is currently in development.]

This content is currently in development.

Chapter 4. Onboarding and Provisioning

This chapter covers the following topics:

• **Configuration Templates:** This section discusses various template types, including CLI, device, and feature templates. This section also covers design and scaling techniques with templates.

• **Developing and Deploying Templates:** This section provides step-by-step instructions for building and deploying device and feature templates.

• **Configuration Groups and Feature Profiles:** This section explains the building blocks of configuration groups as a new way to easily configure devices in Cisco Catalyst SD-WAN.

• **Developing and Deploying Configuration Groups:** This section provides step-by-step instructions for building and deploying configuration groups and feature profiles.

• **Onboarding Devices:** This section covers onboarding of devices with manual setup and automatic provisioning with techniques such as using plug and play (PNP) and a bootstrap configuration file.

• **Key Settings in Device Templates and Configuration Groups:** This section describes some prevalent configuration settings that play important roles in common uses cases, such as with Direct Internet Access breakout,

Technet24

tunnel count scoping, and transport preferences.

Current methods of managing configurations on network devices pose a lot of challenges, including version control, human error, and scaling considerations when deploying to a large number of devices. Traditionally, network engineers make individual changes to various network devices via the CLI. As networks grow, these configurations are often shared or piecemealed with other network devices (such as QoS or routing protocol configurations). Using QoS as an example, many questions must be answered before deployment and, depending on the device, there may be different options for modifying the configuration. Is it MLS queuing or MQC queueing? What hardware platform is it? How many queues has the service provider provided? What DSCP values are you using? This creates a lot of complexity when managing configuration options.

In a perfect world, all of our devices and configurations would be standardized across locations. But this isn't realistic for a multitude of reasons (such as different providers, hardware upgrade cycles, and business needs). As a network grows, the disparity among network devices and network functions makes operations and troubleshooting even more difficult. To compound this issue, network configurations tend to persist as devices are upgraded and replaced. The original intent of the network configuration is likely to get lost as the IT staff turns over or other factors change.

In most cases, network device configurations are rarely revisited for cleanup

or optimization. Network administrators tend to make configuration changes on the fly, and the previous configuration gets lost. Version control is important, especially when there are outages. Having a working configuration to roll back to can save a lot of headache. Oftentimes, outages aren't noticed immediately after a change. With no version control or change management in place, rolling back to the last-known configuration becomes difficult because the previous configuration was not tracked.

With Cisco Catalyst SD-WAN, configuration management is maintained via a robust engine that supports automatic rollback. At this writing, two configuration approaches exist: using templates, which is a widely known configuration method, and using configuration groups, which involves presenting optimizations to the configuration engine.

You can build templates in a modular fashion and reuse them across differing device types. Templates allow network administrators to quickly roll out configurations or changes at a wide scale while also ensuring that the syntax is correct and is supported on any platform.

Thanks to automatic rollback in Cisco Catalyst SD-WAN, if a configuration option gets applied to a device and cuts off the device's ability to be managed by SD-WAN Manager (formerly vManage), the device automatically rolls back after 5 minutes (by default) and allows the network engineer to correct the issue.

Technet24

With today’s networks getting larger and larger, it’s crucial to keep the work of provisioning and onboarding to a minimum. For example, onboarding a network device at a branch involves the following steps:

The device ships from manufacturing to IT staff.

The network administrator applies the required configuration to the device.

The IT staff drives or ships the device to the appropriate location and physically installs it.

This process is expensive for an organization, especially when it needs to occur across thousands of devices. Issues such as a device’s failure to connect to the network add further delay and operational expense as the IT team must perform troubleshooting.

In some cases, device installations are performed remotely. While remote installation reduces the cost of traveling, it can create a lot of frustration when issues arise. With a lack of local IT support at the site, network administrators often end up having to rely on non-IT staff to perform operations on the device. This requires a lot of faith in the remote staff and potentially takes them away from their day job.

Cisco Catalyst SD-WAN simplifies onboarding and provisioning of a device. WAN Edge devices support mechanisms such as PNP and ZTP to automatically bring a device online and into the fabric. These onboarding

solutions work by allowing a network administrator to preconfigure a device within SD-WAN Manager. Once SD-WAN Manager sees the device, it automatically applies the specified configuration. WAN Edge devices can be shipped directly to the remote location and don’t require initial configuration by IT staff. Once a device is physically installed at the remote site, it will automatically locate SD-WAN Control Components and begin the provisioning process. By reducing the time to bring up remote sites, IT staff can bring more devices online more quickly and with fewer errors. This time savings reduces operational cost and allows IT staff to focus on bringing additional capabilities to the business.

Note

The screen captures in this chapter show Catalyst SD-WAN Version 20.12. While newer versions of Catalyst SD-WAN might look slightly different, the basics of the configuration process flow will remain the same.

Configuration Templates

A network administrator can apply configurations in Cisco Catalyst SD-WAN in two different ways:

• They can apply a configuration by using the templates and Configuration Groups in the SD-WAN Manager GUI.

Technet24

• They can programmatically leveraging SD-WAN Manager APIs or configure devices manually via the CLI (that is, by using SSH to connect to a device or by connecting via the console port).

Using the SD-WAN Manager GUI is the preferred mechanism, as it is less error prone and provides support for automatic recovery. Configurations provisioned via SD-WAN Manager can be applied to both WAN Edge devices and SD-WAN control components. When SD-WAN Manager is responsible for applying the configuration, it is the single source of truth, and changes can only be applied via SD-WAN Manager.

Note

To apply centralized policy to the SD-WAN Controller, it needs to have a template assigned and be under the control of SD-WAN Manager. When a component is managed by SD-WAN Manager, an administrator cannot make changes to the device locally.



When applying a configuration to WAN Edge devices or controllers using the SD-WAN Manager GUI, a network administrator can apply a device template to a single device or to multiple devices of the same model. Device templates (see Figure 4-1) can either be CLI based or feature template based.

When a CLI template is being built, the whole configuration must be in the template (not just specific configuration snippets). On the other hand, feature templates can be thought of as building blocks wherein each block is a specific technology feature.

A Feature Template defines the specific feature or technology to enable or configure, such as routing protocols, interface parameters, and Overlay Management Protocol (OMP). Feature templates can be reused between multiple device templates, and this flexibility brings greater scale to the solution (which is why using feature templates is the recommended way of configuring devices). Feature templates can be device type agnostic as well.

A network administrator needs only to be concerned with the functionality of the WAN edge device. SD-WAN Manager is responsible for converting a template to a device’s corresponding configuration syntax for the appropriate OS and platform.

Technet24

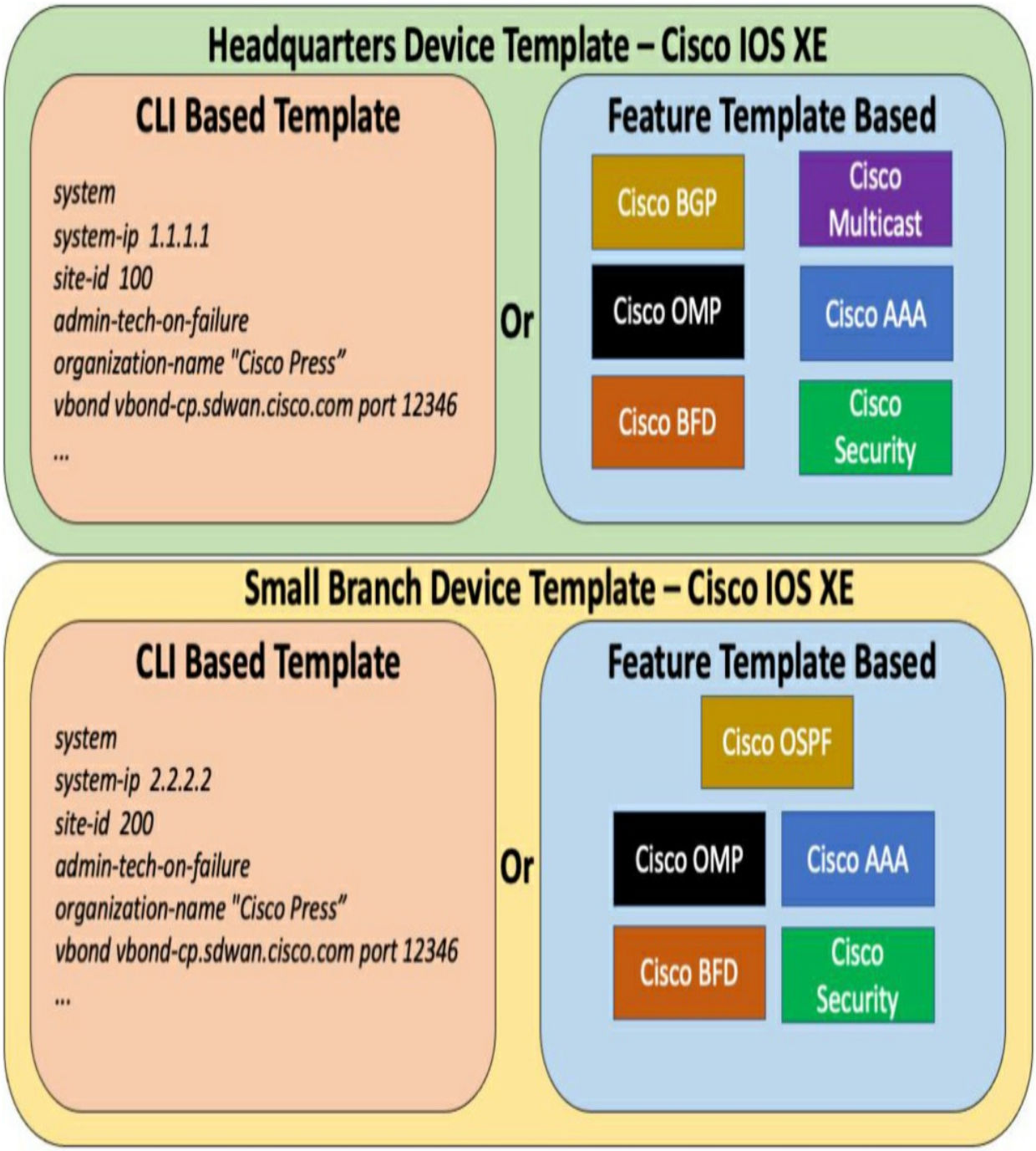


Figure 4-1 *Device Templates*

A Device Template is a collection of feature templates and can only be applied to specific device types. For this reason, you may have multiple device templates for the same model of hardware, depending on a particular

device’s location, connectivity options, or role in the network.

Note

A device template can’t be shared across different device types, but a feature template can be used across multiple different device types. When a portion of the configuration from a device template needs to be reused, a template copy can be performed.

As illustrated in Figure 4-2, a device template has four main parts:

Technet24

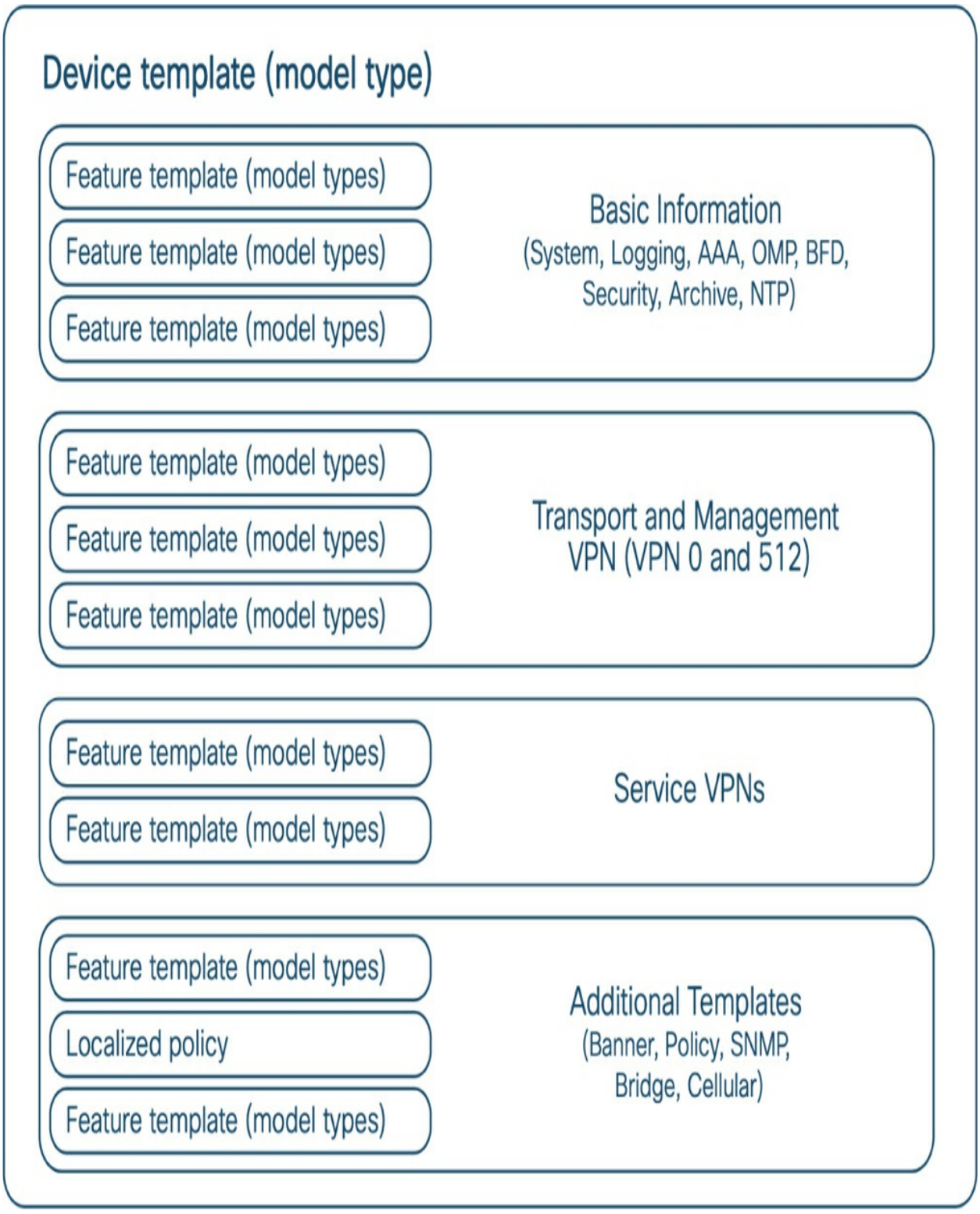


Figure 4-2 *Device Template Structure*

• **Basic Information:** This section includes items such as system, logging, AAA, BFD, and OMP feature templates.

• **Transport and Management VPN:** This section has templates for configuration of VPN 0 and VPN 512 (such as underlay routing protocol configuration and interface configuration).

• **Service VPN:** This section is where service VPNs are defined and LAN-facing template configurations. It is where BGP, OSPF, and interface parameters are configured.

• **Additional Templates:** This section is for local policies, security policies, CLI add-on templates, ThousandEyes Agent, SNMP configuration templates, and so on.



Feature Templates provide flexibility in configuration options. For example, feature templates provide the option to define variables for configuration parameters. This allows you to reduce the number of templates required in your deployment yet make the deployment much more modular. For example, suppose you have MPLS transports that use different physical interface numbers: GigabitEthernet0/0/0, GigabitEthernet0/0/1, GigabitEthernet0/0/2, and so on. Initially, you might want to build a feature template for each physical interface that has a different IP address. This would result in three different templates. By using variables for the physical

Technet24

interface and IP address options, you could condense this down to one feature template that can be used across all device templates.

Figure 4-3 shows how variables can be used to control template sprawl. This example involves nine different interface templates, and the one that is used depends on whether the IP address is assigned via DHCP and what interface is used. By using variables, a network administrator can reduce this down to three different feature templates.

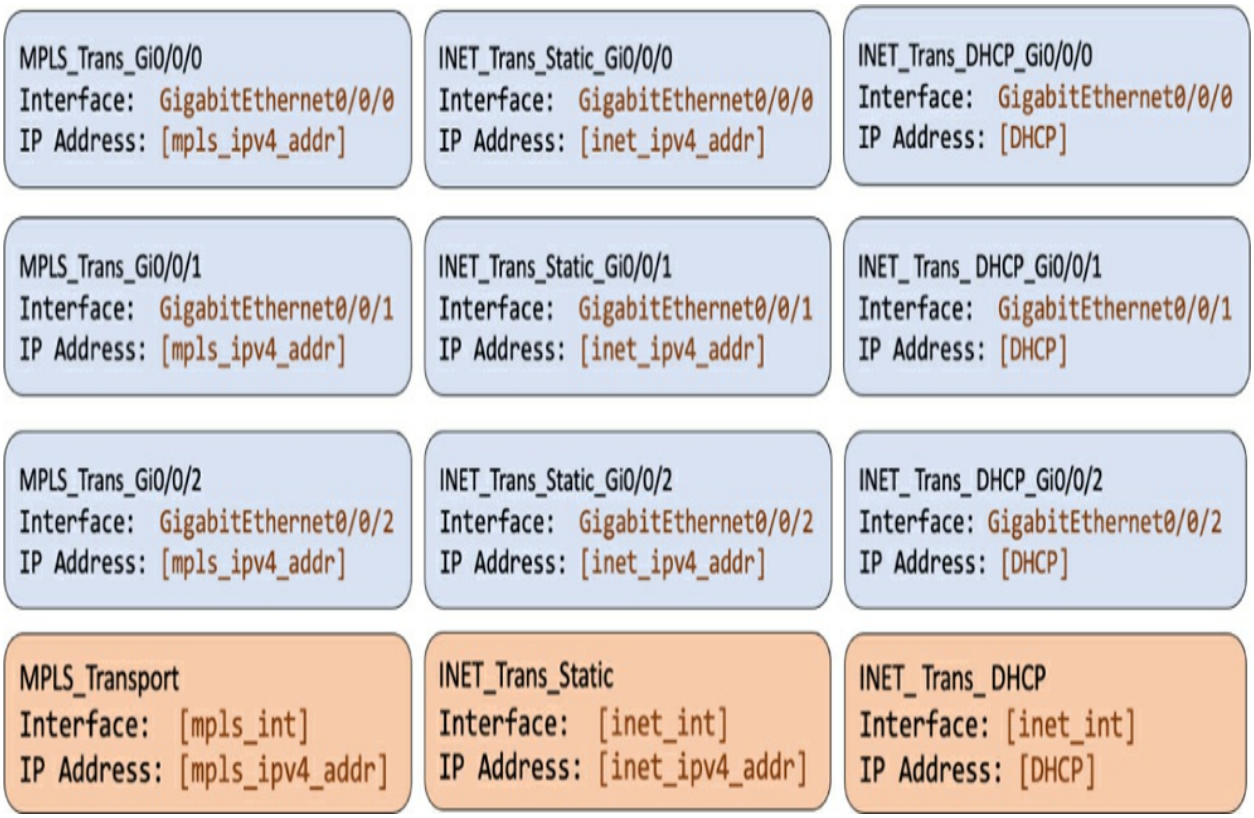


Figure 4-3 *Controlling Template Sprawl*

Three types of values can be defined in a template:

• **Default:** This is the factory default value, and it cannot be changed. An

example might be using the default BFD timers.

• **Global:** Values set here are the same wherever this configuration option is used. An example could be SNMP community strings that you want to be globally applied to all devices using this template. The beauty of this is that, later on, if there needs to be a change to these values, you just update the feature template global option, and it updates every device template that is using this feature template.

• **Device specific:** The value is set via a user-defined variable (as in our earlier example with interface names). The values for these variables are set when the device template is attached to a specific device.

In Figure 4-4, you can see how these referenced values can be used. Some template options might not have all three of these options, depending on what is being configured. For example, a BGP AS number won't have a default value.

Technet24

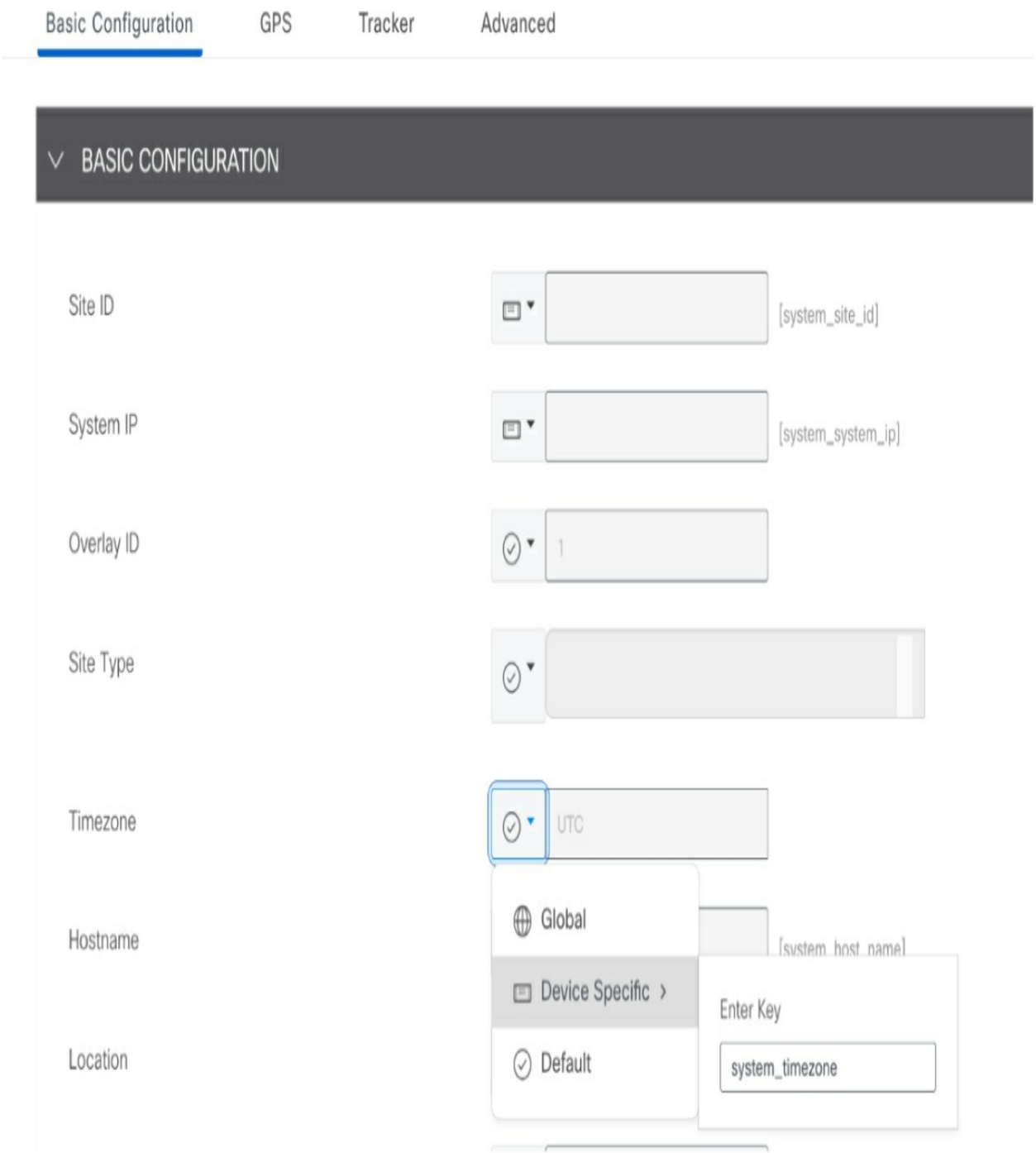


Figure 4-4 *Setting Variables*

A large number of feature template options can be configured. Here are some common feature templates:

• **System:** You can configure basic system information such as system IP address, site ID, and hostname.

• **BFD:** You can adjust BFD timers and app route multipliers for each transport or color. BFD timers are used for application-aware routing.

• **OMP:** You can change graceful restart timers or control redistribution from other routing protocols into OMP.

• **Security:** You can change IPsec security settings such as anti-replay, authentication, and encryption settings.

• **VPN:** You can define a service VPN, routing protocol redistribution, or static routing.

• **BGP:** You can configure BGP in a VPN or VRF instance.

• **OSPF:** You can configure OSPF in a VPN or VRF instance.

• **VPN interface:** You can define an interface that is part of a service VPN or VRF instance. Common configuration options include IP address, QoS, ACLs, and NAT.

As feature templates are defined, they can be referenced via a device template. The device template can be applied to a specific device or to a group of devices. Remember that a device template can only be built for a specific device type. If there are any variables defined in the feature template,

Technet24

when the device template is attached, these values will need to be populated. Once these values are defined, a configuration syntax check is done in SD-WAN Manager. If it is successful, SD-WAN Manager notifies the WAN Edge device regarding the configuration that needs to be retrieved, and the device pulls it. Feature template variable values can be populated either within the SD-WAN Manager template attachment workflow or by using a CSV file. Populating feature template variables via CSV allows an administrator to quickly provision many devices at once.

If the need arises to change these values after the device template has been applied, it is possible to change these values on a device-by-device basis. If any changes are made to a feature template or device template (for example, changing the IP address of a device interface, changing the username and password of a device), SD-WAN Manager immediately pushes the updated configuration to all devices using that template.

Note

If at the time the configuration is pulled by the device, it loses control plane connectivity to SD-WAN Manager, by default the WAN Edge device starts a rollback timer of 5 minutes. If it doesn’t reestablish connectivity within that 5 minutes, it rolls back its configuration and reconnects to SD-WAN Manager using the last-known-good configuration. At that time, the network administrator will see that the device is out of sync

and can correct the issue.

Developing and Deploying Templates

Template configuration and creation is performed in the SD-WAN Manager GUI. After an SD-WAN Manager instance is deployed, some default templates are created. These templates can be used as a starting point to create customized templates. For example, to create a BFD template, you follow these steps:

Navigate to **Configuration > Templates** and go to the configuration section for templates, as illustrated in Figure 4-5.

Technet24

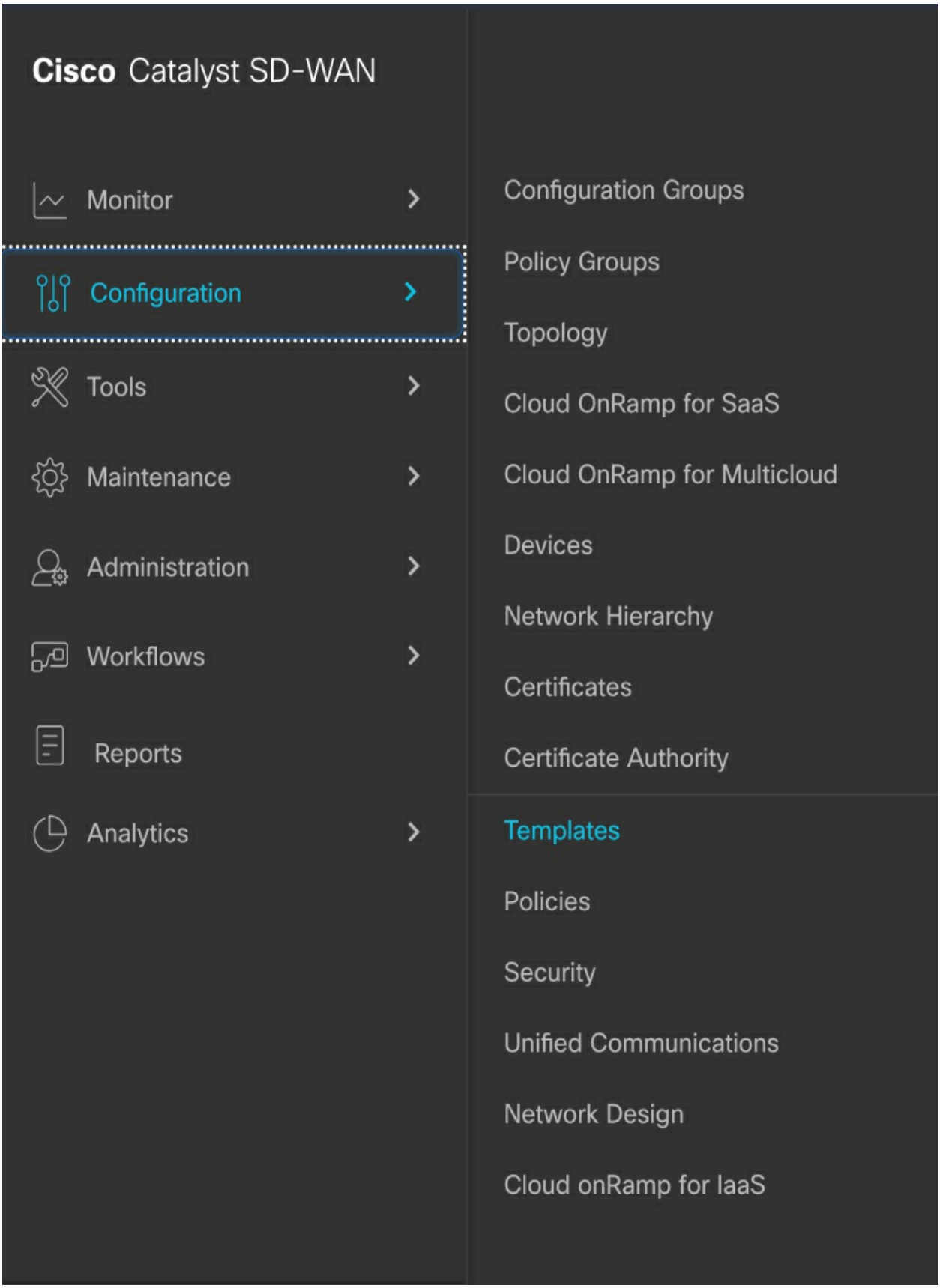

Cisco Catalyst SD-WAN
Monitor
Configuration
Tools
Maintenance
Administration
Workflows
Reports
Analytics
Configuration Groups
Policy Groups
Topology
Cloud OnRamp for SaaS
Cloud OnRamp for Multicloud
Devices
Network Hierarchy
Certificates
Certificate Authority
Templates
Policies
Security
Unified Communications
Network Design
Cloud onRamp for IaaS

Figure 4-5 *Accessing the Template Configuration Interface*

In the template configuration window, you're presented with the option to configure device templates or feature templates, as shown in Figure 4-6.

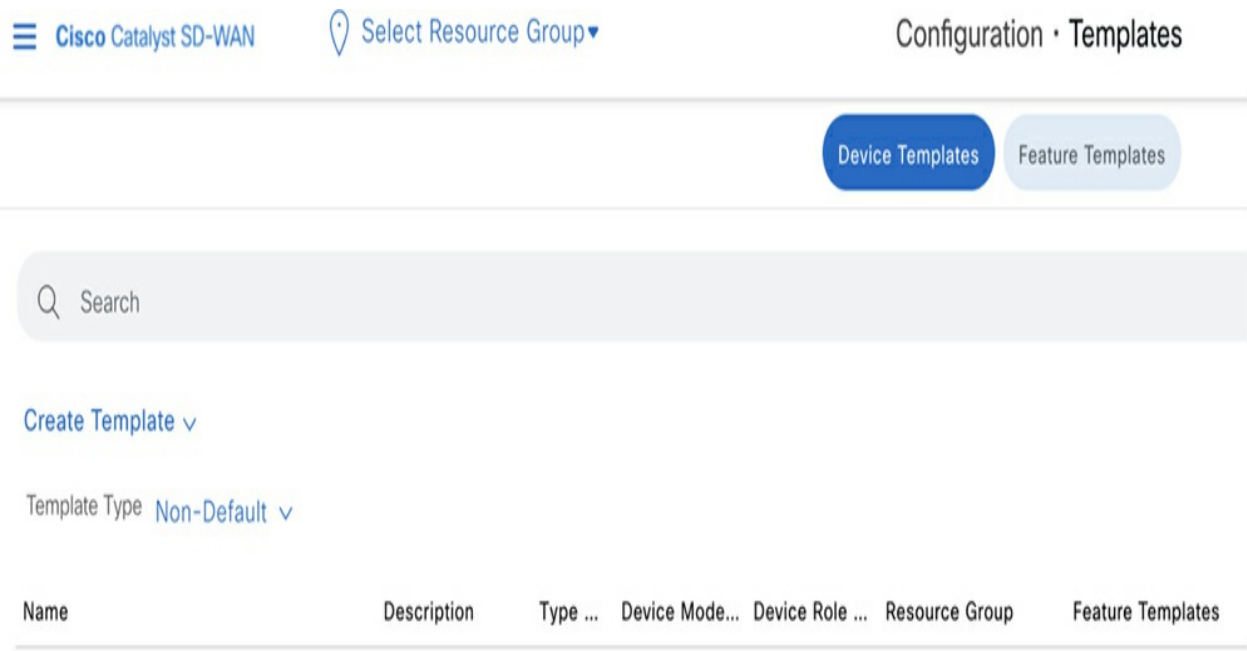


Figure 4-6 *Template Configuration Window*

To begin creating a feature template, select **Feature Templates** and click **Add Template**. Select the devices that this template will apply to and select the type of template, as shown in Figure 4-7.

Technet24

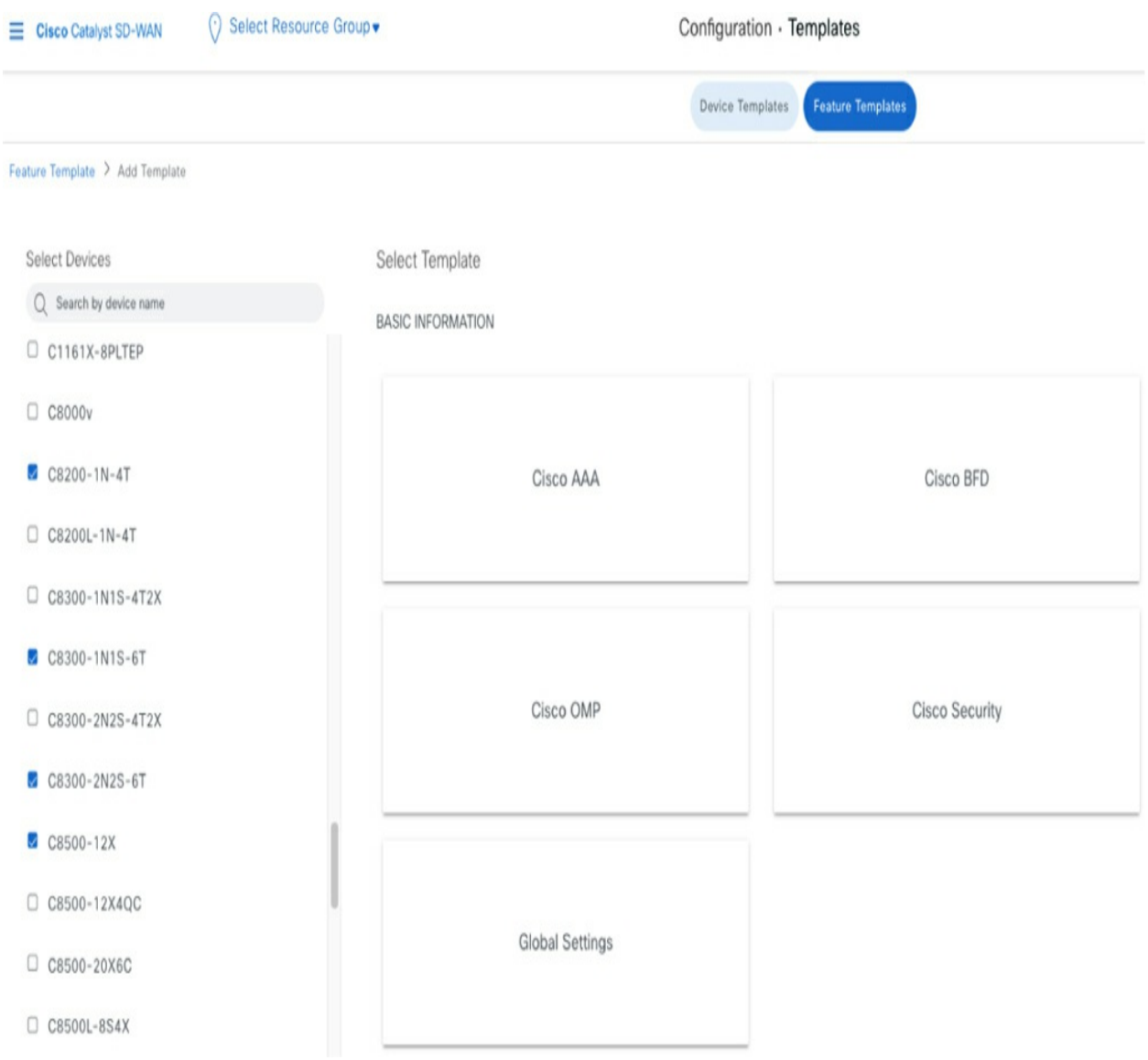


Figure 4-7 *Template Configuration: Device Selection Window*

Start setting values, as shown in Figure 4-8. These values can be either variables, global parameters, or the default parameter. When you're done setting configuration options, name the template **Sample-BFD-Template** and click **Save**.

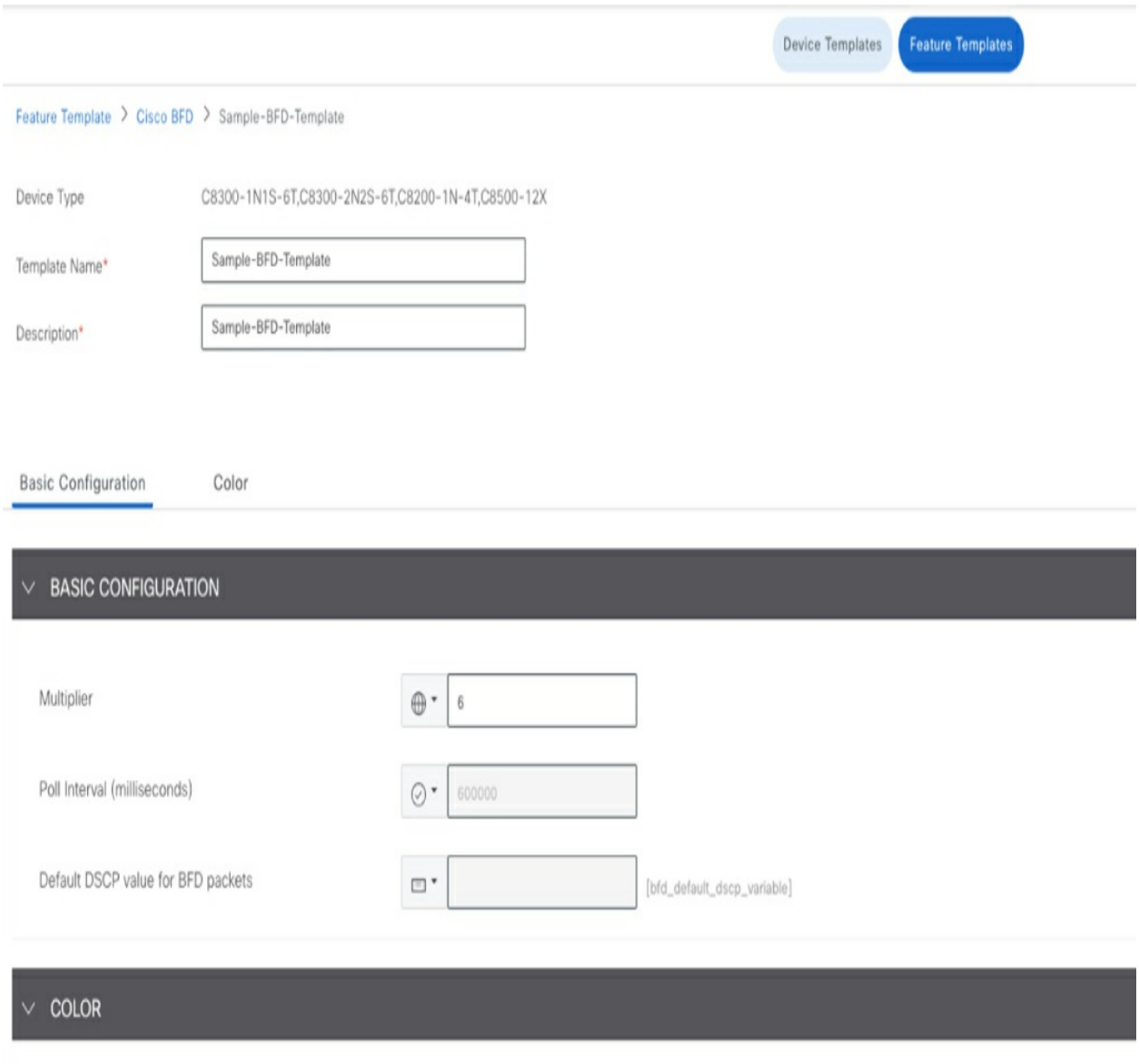


Figure 4-8 *Template Configuration: Setting Configuration Values*

To attach a feature template to a new device template, click **Device Templates** and then **Create Template**. (Note that you can also attach a feature template to existing device templates.) When you see the option to create a CLI-based template or a feature-based template, select **From Feature Template**, as shown in Figure 4-9.

Technet24

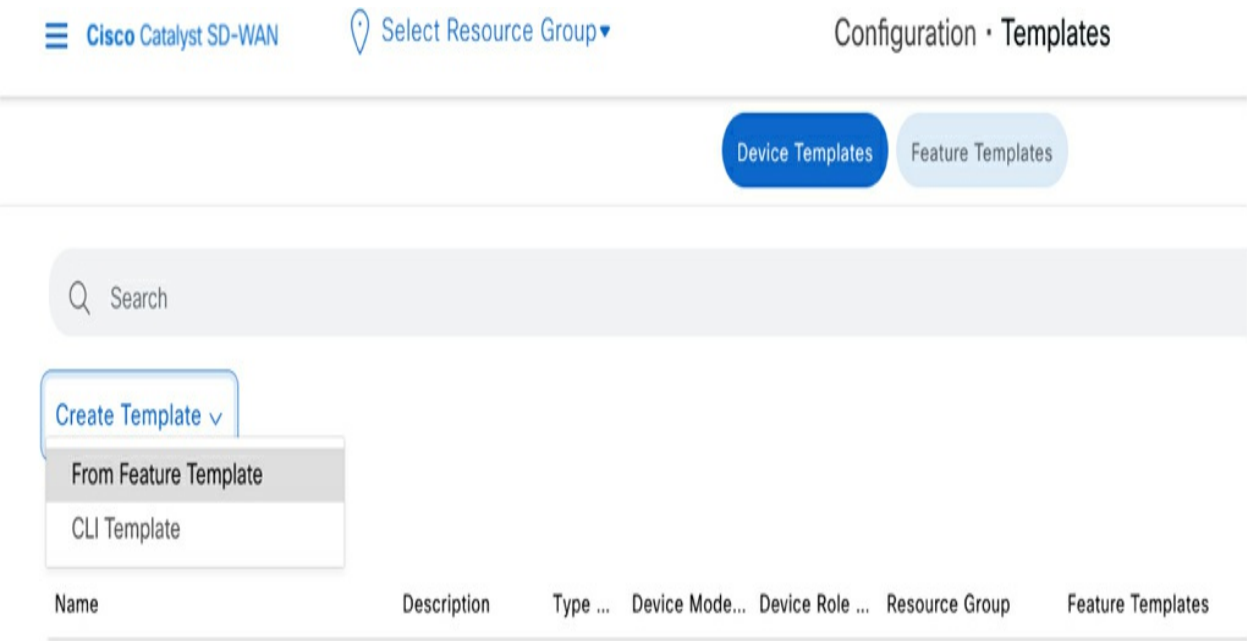


Figure 4-9 *Template Configuration: Selecting a Device Template*

Select the device model this template will apply to, select the device role, name the template **HQ-Device_Template**, and provide a description for the template, as shown in Figure 4-10.

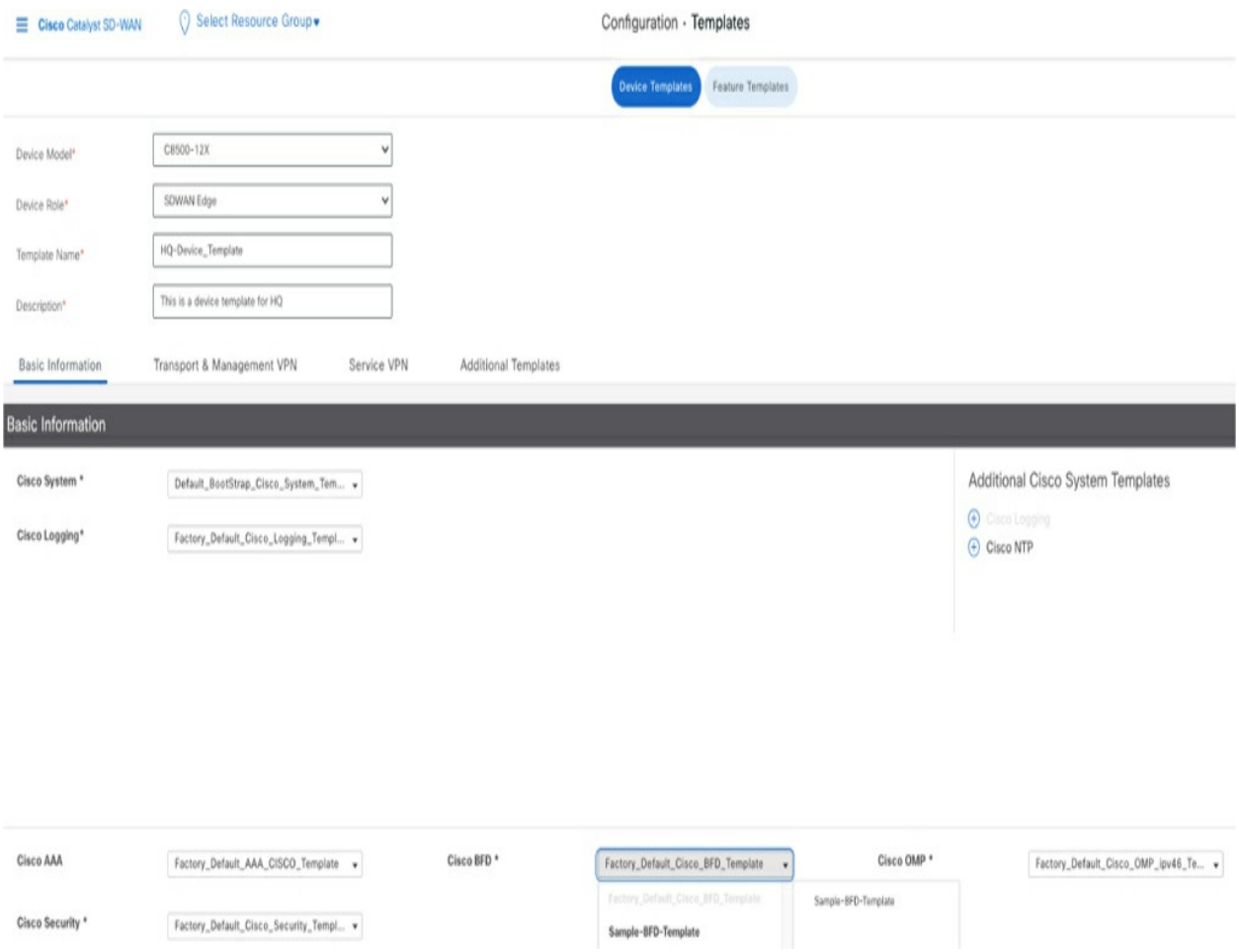


Figure 4-10 *Template Configuration: Selecting a Device Feature Template*

To select the feature template to use, select the **Sample-BFD-Template** template you created earlier and click **Save**.

To attach the device template to devices, on the Device Templates page, click the three dots next to the template you wish to attach. Select the option to **Attach Devices**, as shown in Figure 4-11.

Technet24

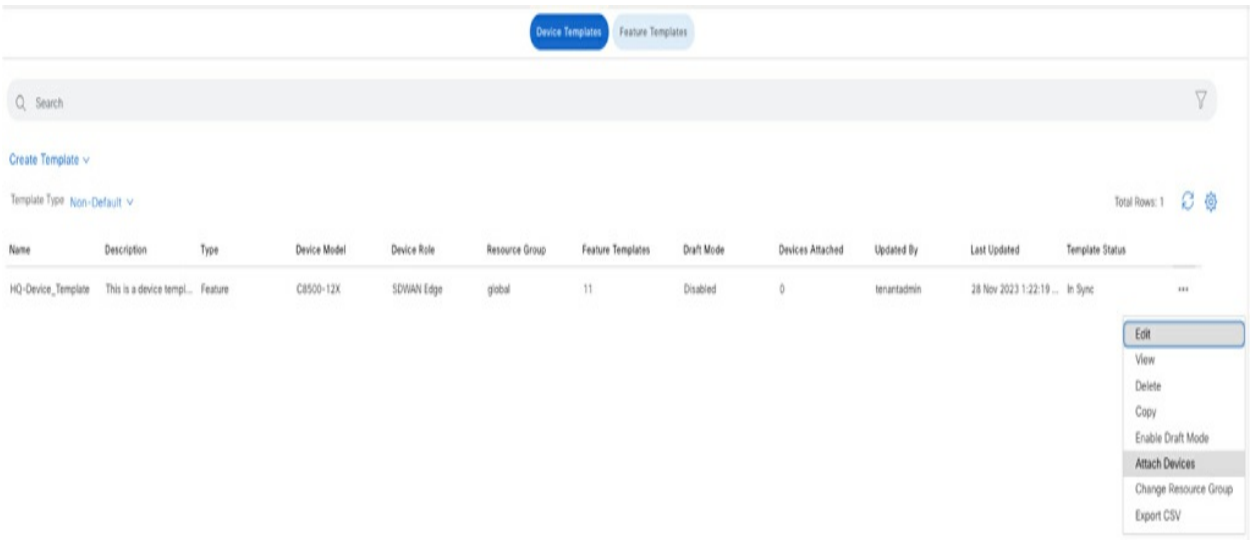


Figure 4-11 *Attaching a Template to a Device*

Select the devices to apply the configurations to, as shown in Figure 4-12. Once you select a device, you have the option to populate any variables.

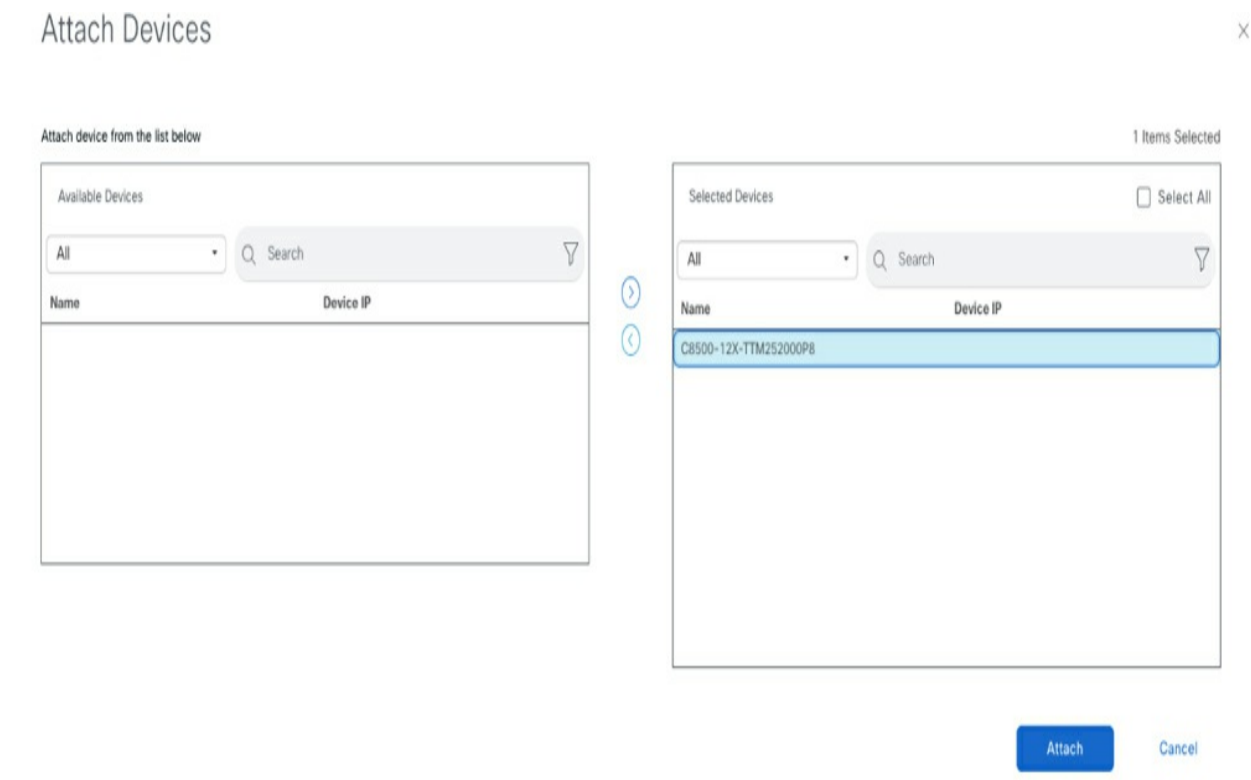


Figure 4-12 *Device Selection Window*

Once you have provided the values of the variables, you can deploy the configurations to the selected WAN Edge devices.

Configuration Groups and Feature Profiles

Starting with Catalyst SD-WAN Version 20.8, which provides backward compatibility with Version 17.6 for WAN Edge devices, Cisco introduced a new way to configure WAN Edge devices, adding benefits such as simplicity through intuitive and sequential configuration workflows, device-agnostic components that permit reusability, and structure that allows you to group devices that share common settings. You don't have to create a separate template for every device type, as you do with device templates. As illustrated in Figure 4-13, Configuration Groups method is composed by three main components:

• **Configuration Groups:** A Configuration Group is a logical grouping of devices or set of configurations that share a common purpose within the WAN fabric. These groups are customizable based on business needs, such as region (for example, East, West) or type of site (for example, retail store, distribution center).

• **Feature Profiles:** A Feature Profile is a flexible configuration bucket based on required similar features, such as WAN or LAN features.

Technet24

• **Feature Parcels:** Feature Parcels are the building blocks of feature profiles. They are individual capabilities that can be used across Feature Profiles.

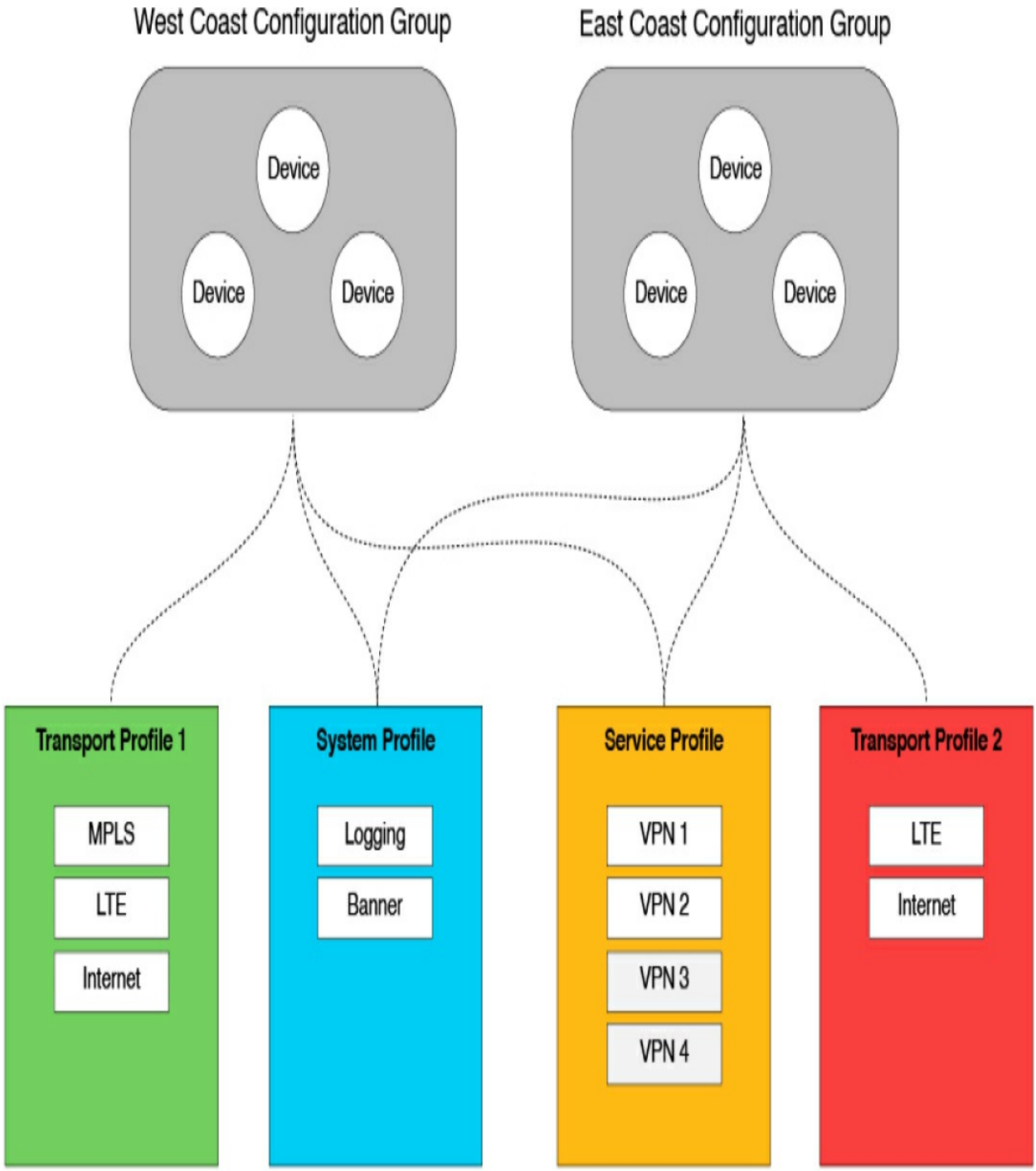


Figure 4-13 *Configuration Groups, Feature Profiles, and Feature Parcels*

Example

In Figure 4-13, the West Coast and East Coast Configuration Groups cater to different business needs, based on different regions. The figure also shows as examples of Feature Profiles (Transport Profiles 1 and 2, System Profile, and Service Profile) that group individual functions, called Feature Parcels, like logging and WAN transport interfaces (such as MPLS or LTE).

Note

You can associate a device to either a configuration group or a device template but not both.

Developing and Deploying Configuration Groups



Figure 4-14 shows two ways exist to create Configuration Groups:

• Select Configuration > Configuration Groups > Add Configuration Group

• Select Workflows > Workflows Library

Note

Remember that the screen captures in this chapter show Catalyst SD-WAN Version 20.12.

Technet24

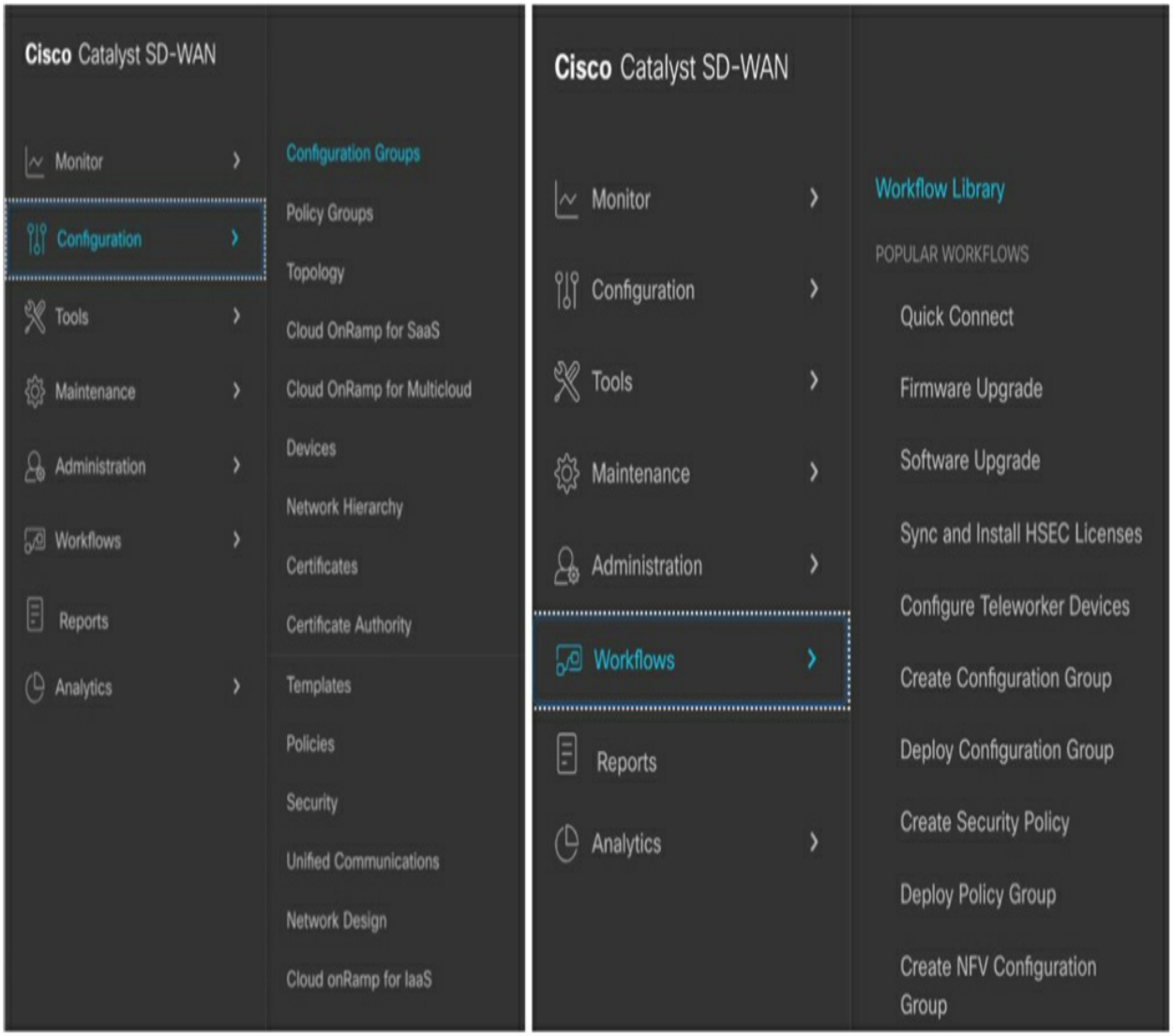


Figure 4-14 *Accessing the Configuration Groups Workflow Interface*

Regardless of the path previously taken, you need to select Create Configuration Group, as shown in Figure 4-15. After the previous selection, the window seen in Figure 4-16 pops up, where you click Let's Do It. Then, you follow these simple steps to complete the configuration group creation process:

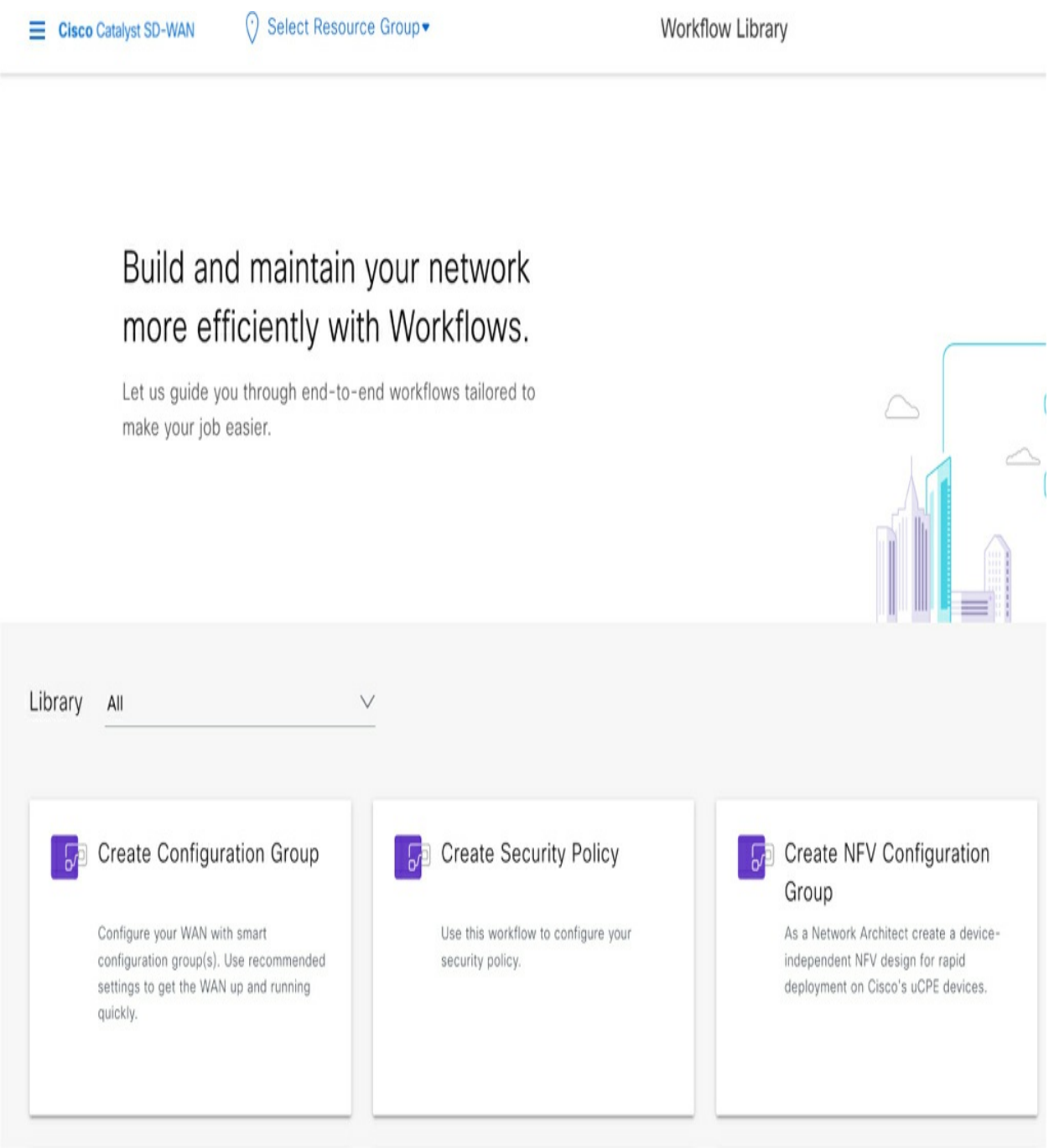


Figure 4-15 *Accessing the Configuration Groups Workflow Interface*

Technet24

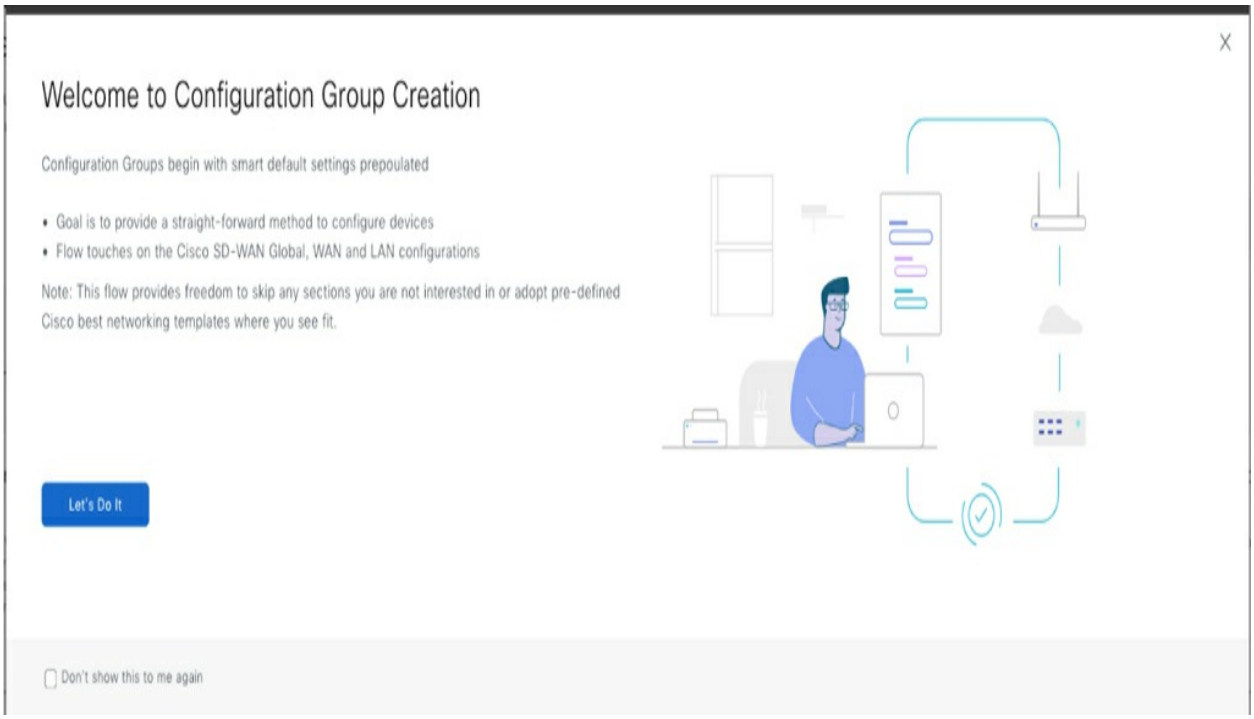


Figure 4-16 *Configuration Group Default Welcome Screen*

Name the Configuration Group and provide a description for it, as shown in Figure 4-17. Then click **Next**.

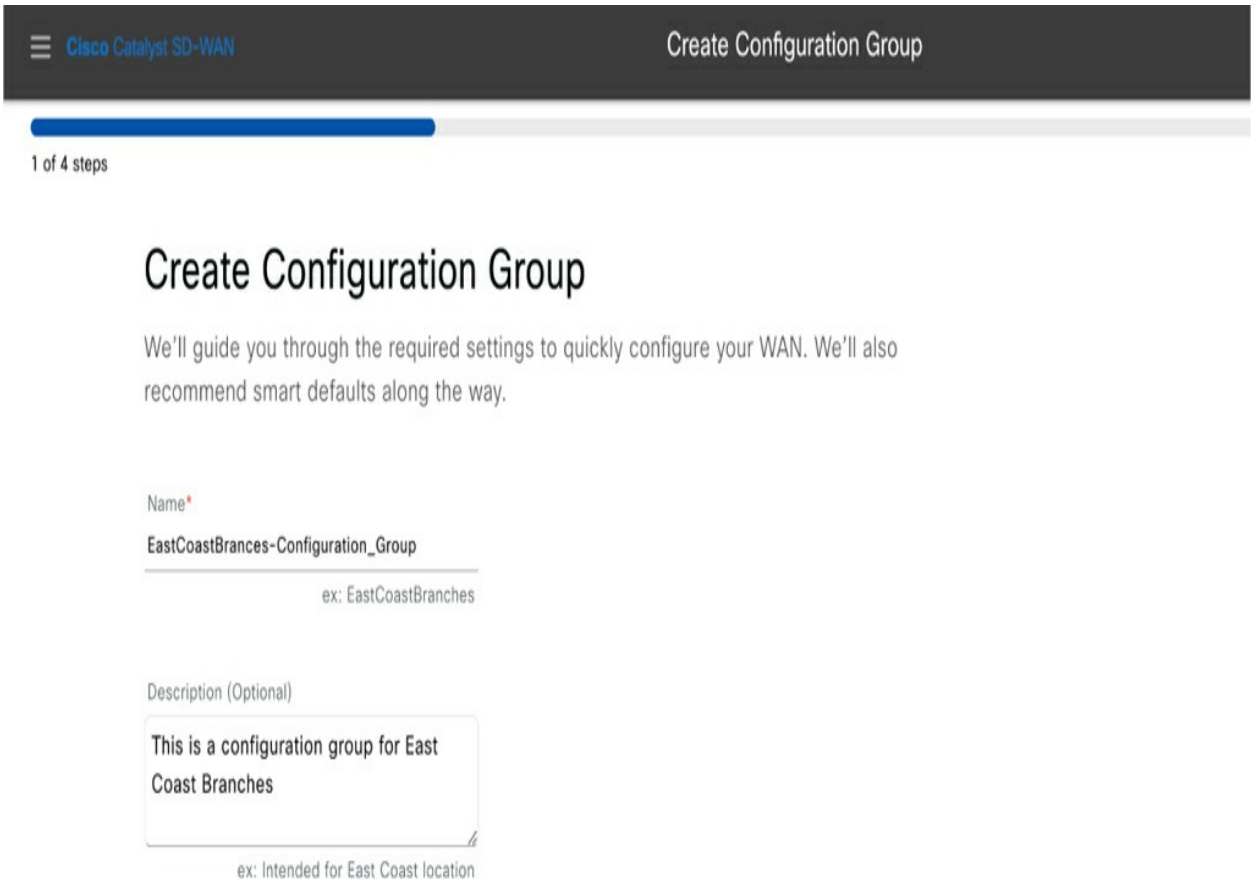


Figure 4-17 *Naming the Configuration Group*

On the Site Configurations screen that appears, select **Single Router** or **Dual Router** under in Site Type, depending on your configuration needs. Under Site Settings, choose the appropriate option, depending on whether you are working with global parameters, variables, or default parameters. Figure 4-18 shows these two first settings you can configure.

Technet24

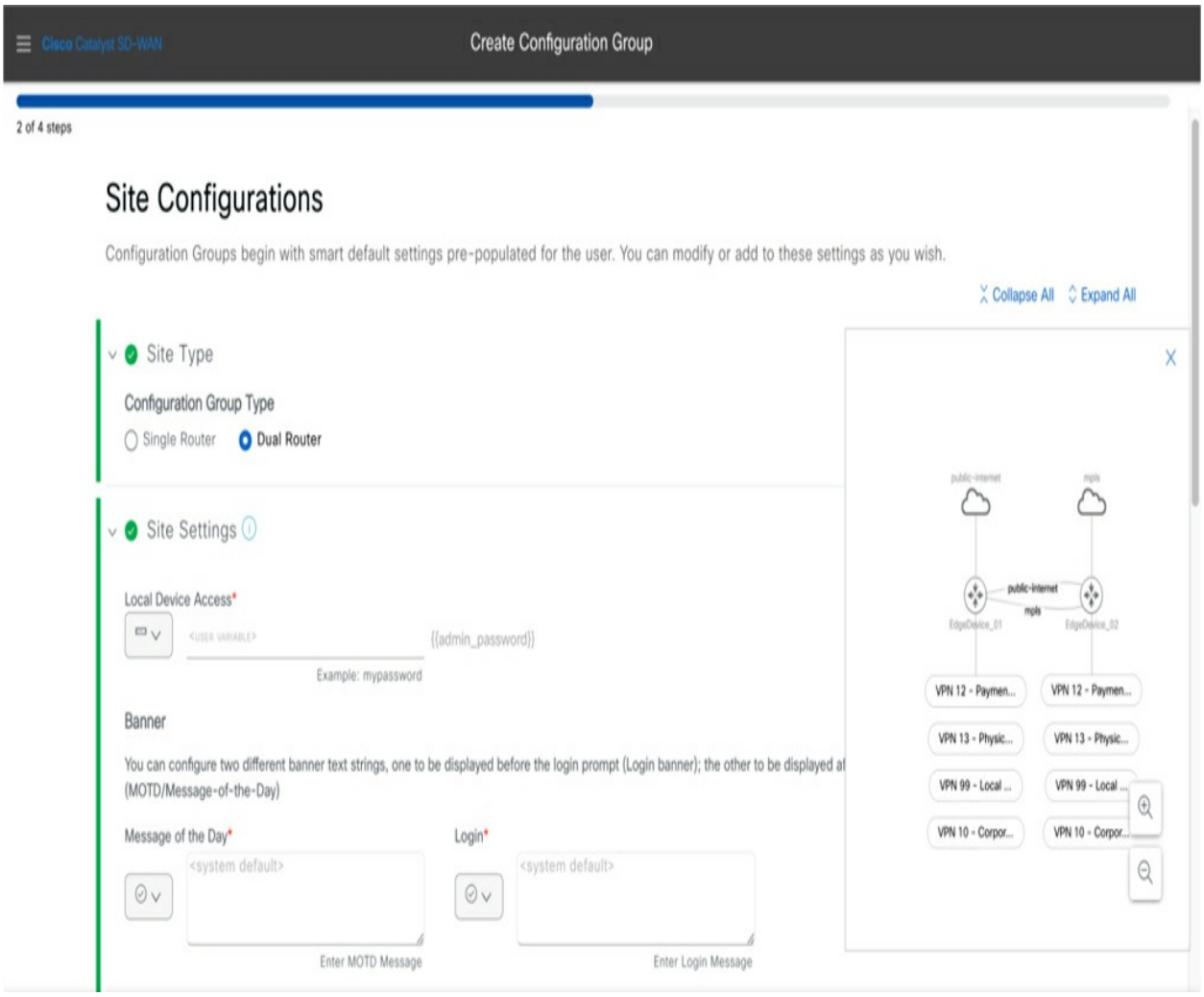


Figure 4-18 *Site Configurations*

Under WAN Interfaces, select the required WAN transports—either **DHCP** or **Static**. Full Mesh or Transport Extension appears as an option only if the Site Type selection is Dual Router. WAN routing and advanced capabilities are also option, as shown in Figure 4-19. This page presents a similar selection for a LAN and Service VPN profile. Click **Next**.

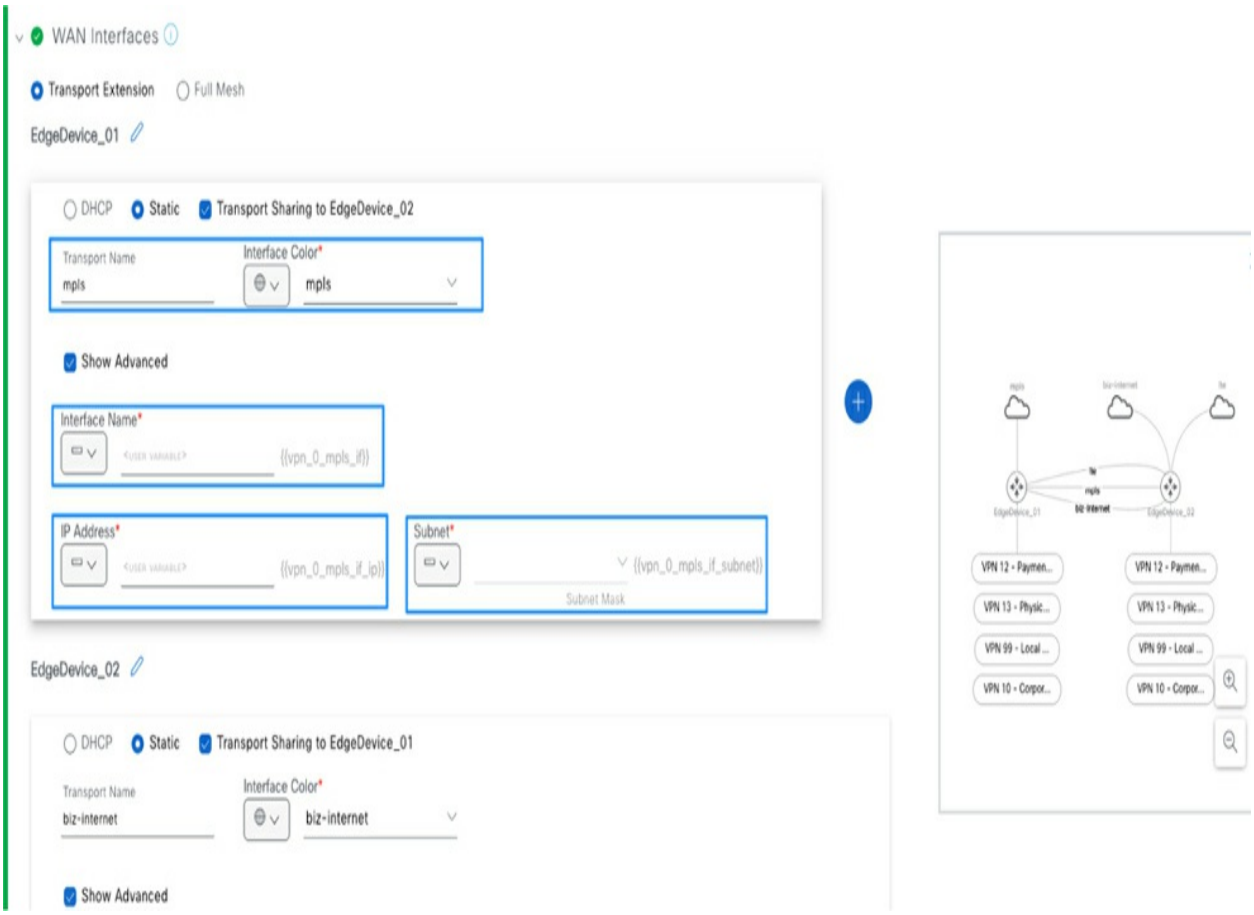


Figure 4-19 *WAN Interface Settings*

Note

Variables provide intelligent and intuitive names related to the interface's color, as you can see in the Show Advanced section in Figure 4-19. You can opt to leave default values to edit later in the process, as required.

On the screen that appears next, set additional features such as Remote access functionality if desired (see Figure 4-20).

Technet24

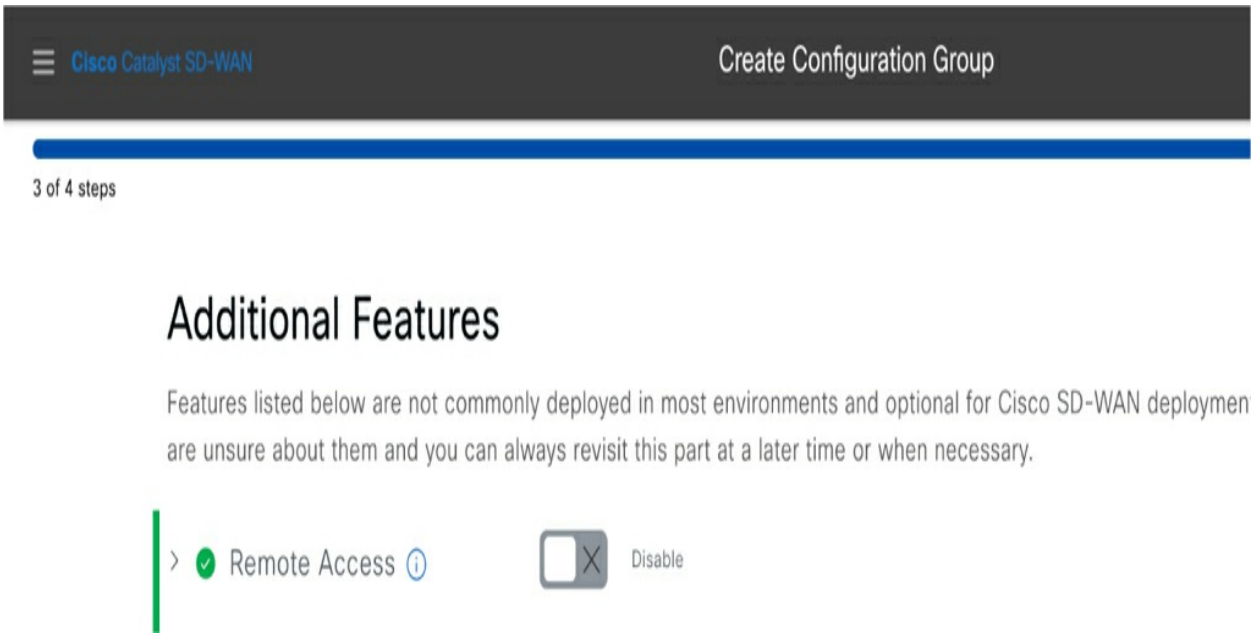


Figure 4-20 *Additional Features Step*

Note

Figure 4-20 shows that the only additional feature available is Remote Access. Recall that the figures in this chapter show SD-WAN Version 20.12, and newer releases may provide different options.

Review the summary that is provided and click **Create Configuration Group** (see Figure 4-21).

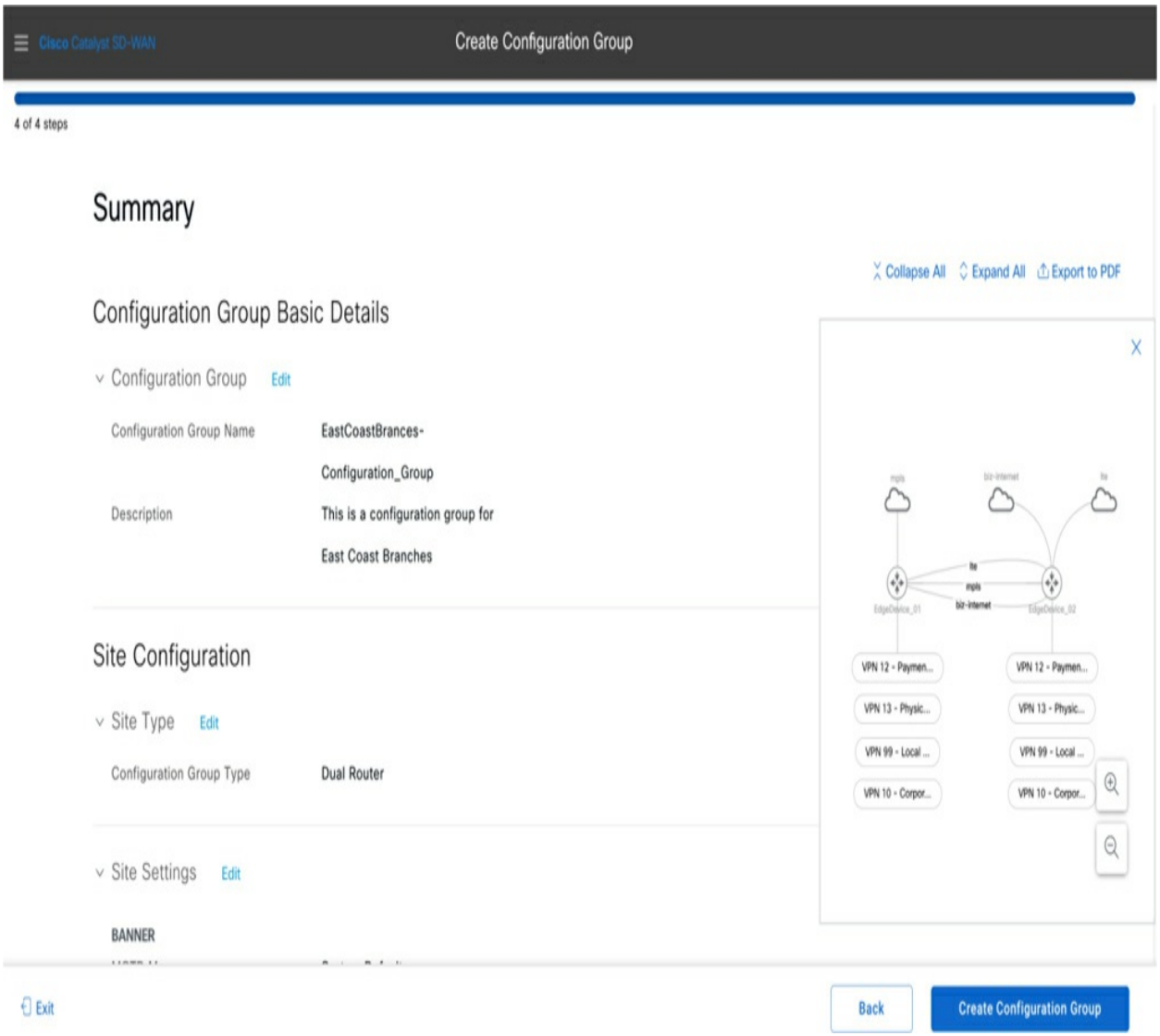


Figure 4-21 *Configuration Group Summary Screen*

Immediately after creating a Configuration Group, you need to associate this group to one or more devices, as shown in Figure 4-22

Technet24

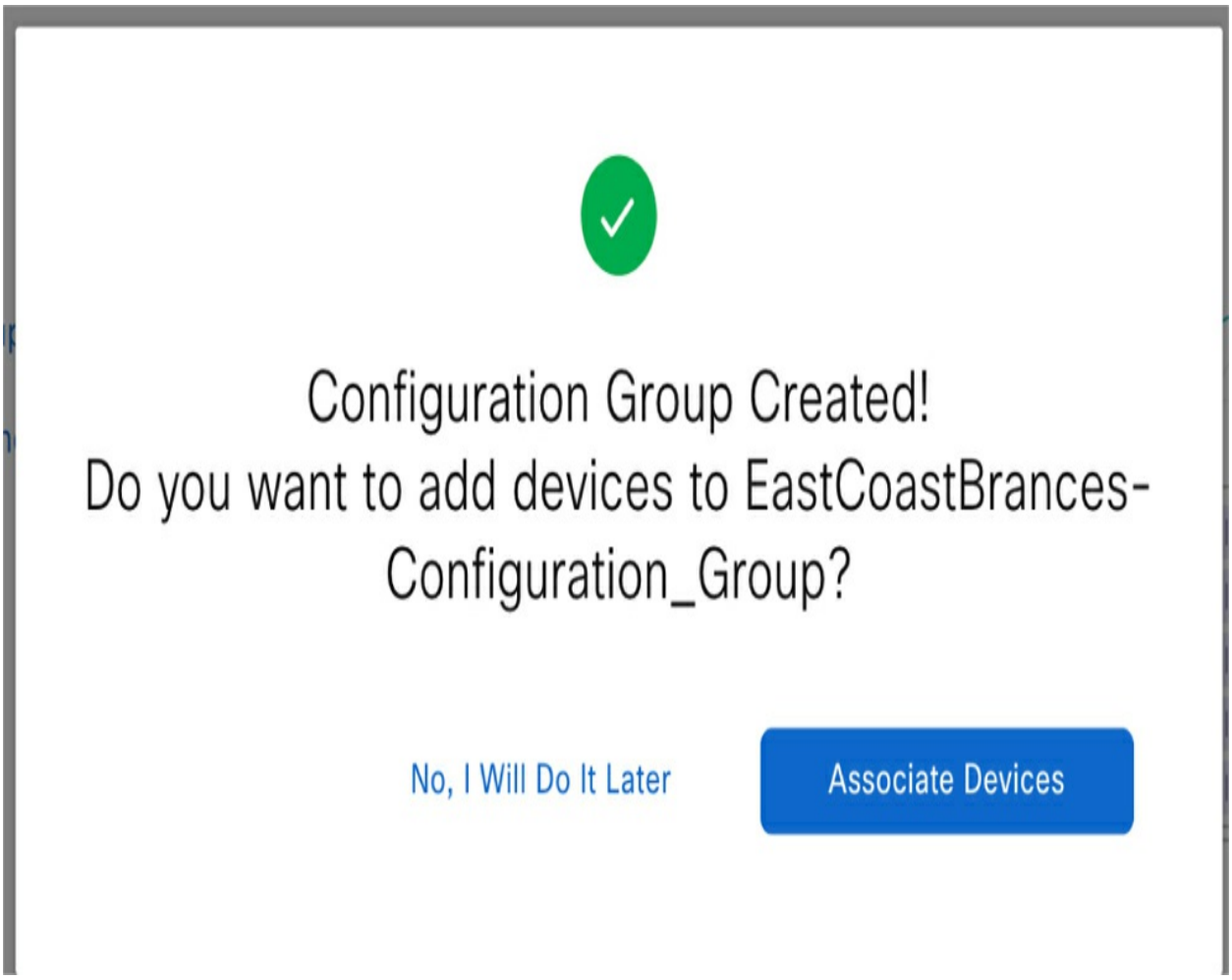


Figure 4-22 *Associate Devices Initial Workflow*

To associate a device to the recently created logical grouping, click **Associate Devices** in the popup window and follow these steps:

On the first screen, click **Next** (see Figure 4-23).

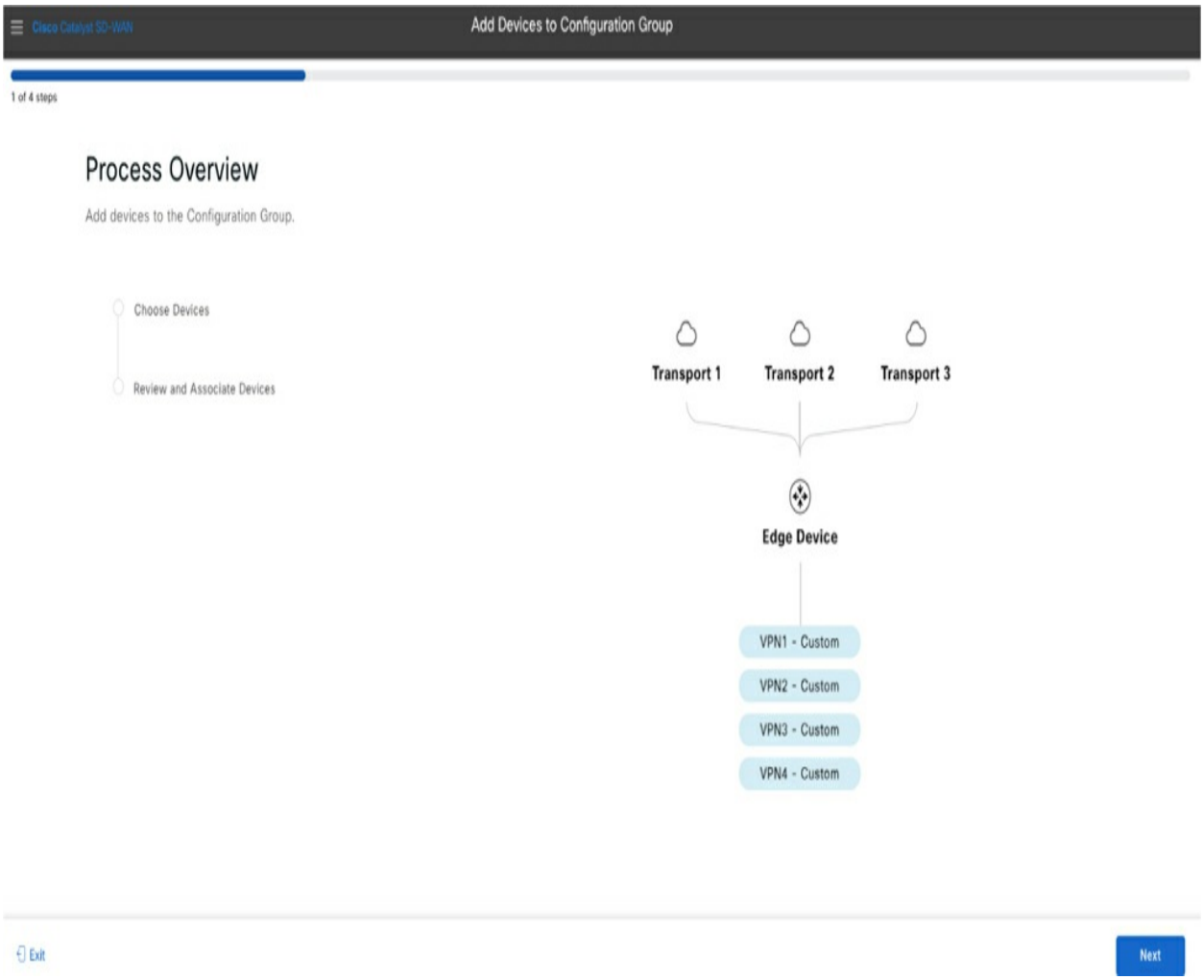


Figure 4-23 *Adding Devices to the Configuration Group Initial Flow*

On the screen that appears, select devices from the list of available devices and click **Next** (see Figure 4-24).

Technet24

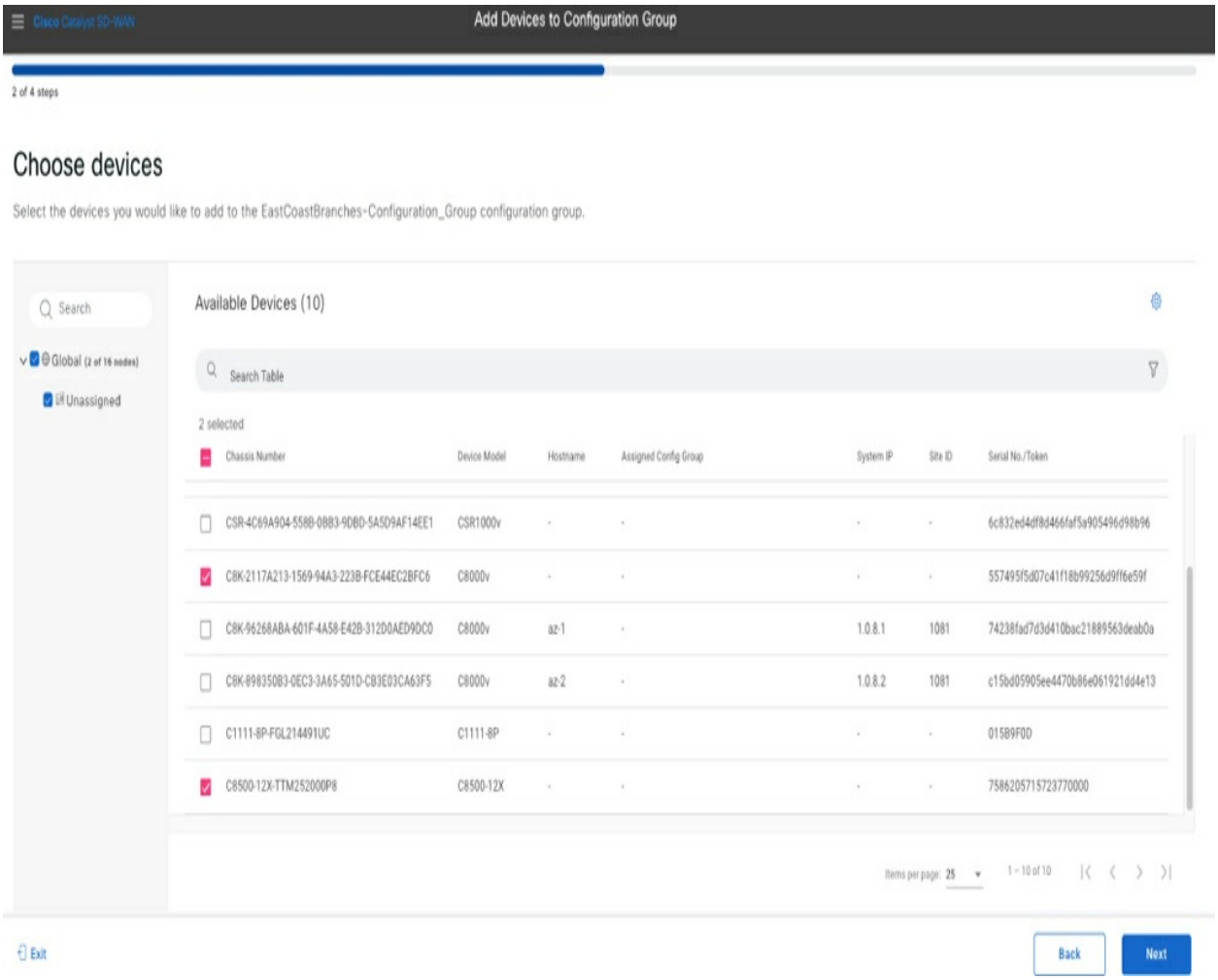


Figure 4-24 *Selecting Available Devices*

On the screen that appears, select the tags to use for grouping and describing the devices (see Figure 4-25). This makes it easier for operations team in the future to differentiate the devices when a Dual Router Site Type selection is done. SD-WAN Manager creates smart default values—in this case, tag names—that can be modified based on needs. When you're finished with your selections, click **Next**.

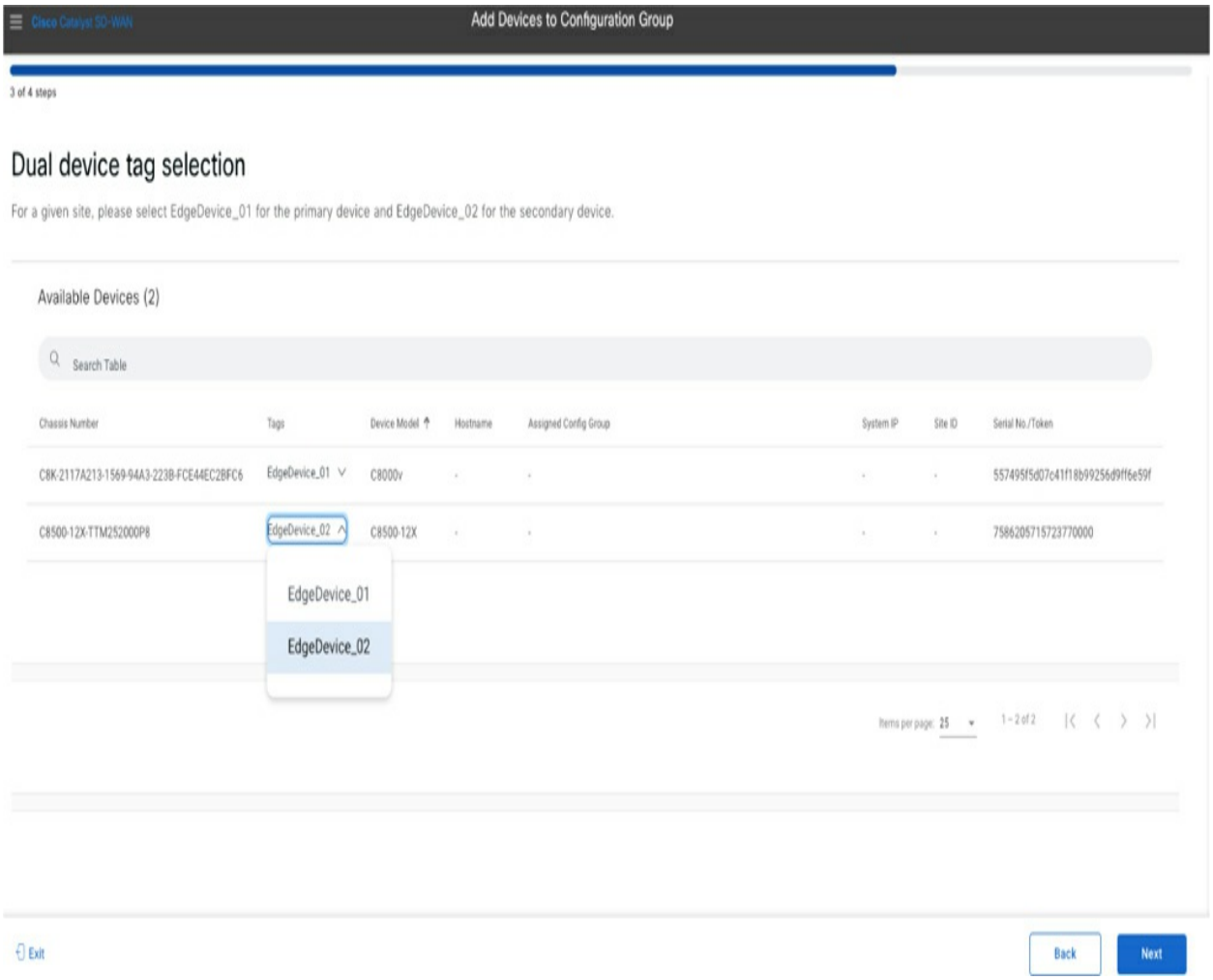


Figure 4-25 *Selecting Device Tags*

On the summary screen, which shows the devices to associate with the Configuration Group and allows you to edit the list to add devices, if everything looks good, click **Save** (see Figure 4-26).

Technet24

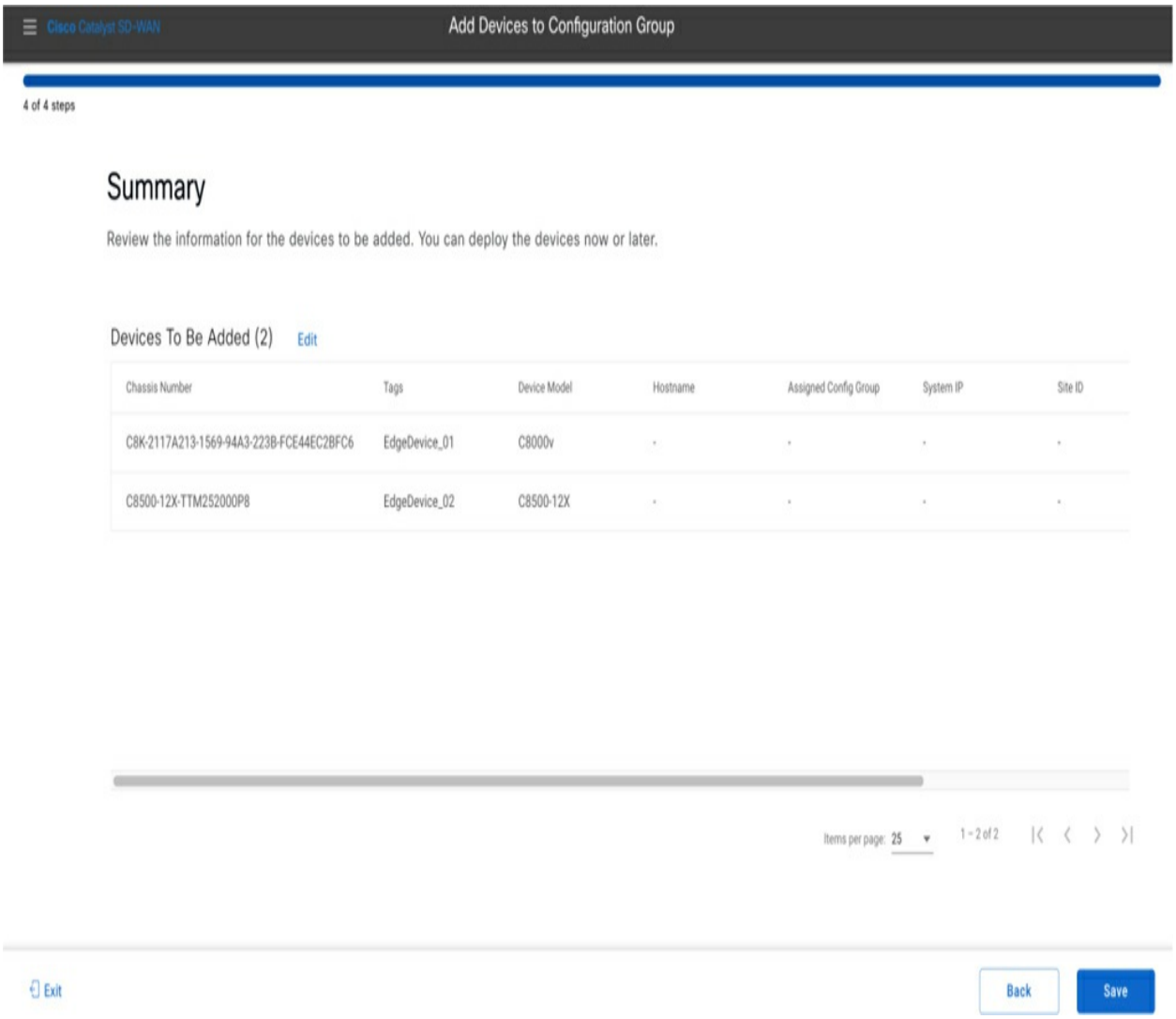


Figure 4-26 *Summary of Devices to Be Added to the Configuration Group*

Next, you have the option of provisioning devices with the Configuration Group you just created or selecting to provision it later, as shown in Figure 4-27. Select **Provision Devices** and then follow these steps:

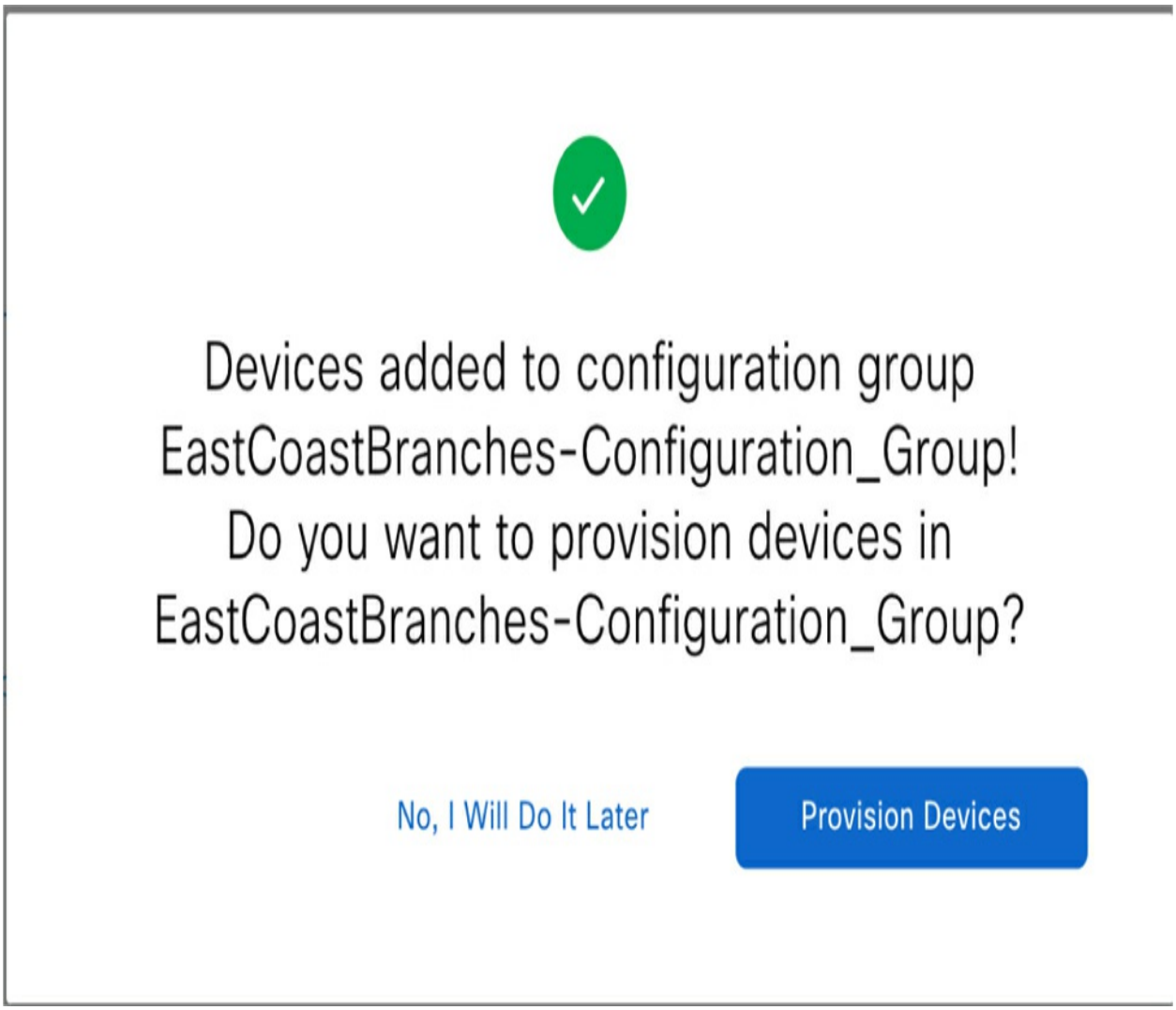


Figure 4-27 *Provisioning Step Prompt*

Take a look at the workflow shown in Figure 4-28 and click **Next**.

Technet24

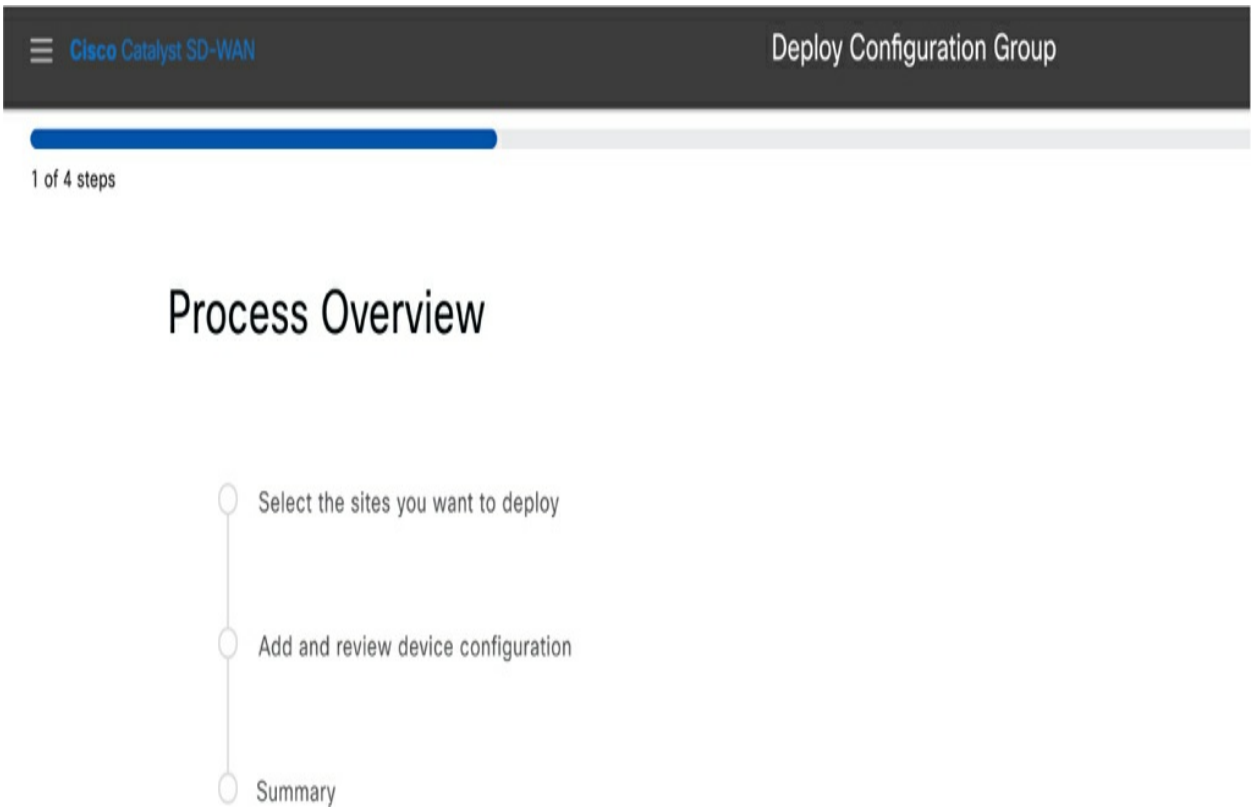


Figure 4-28 *Configuration Group Deployment Initial Screen*

To select the sites to deploy, check the boxes for both **EdgeDevice_01** and **EdgeDevice_02**, as shown in Figure 4-29, and click **Next**.

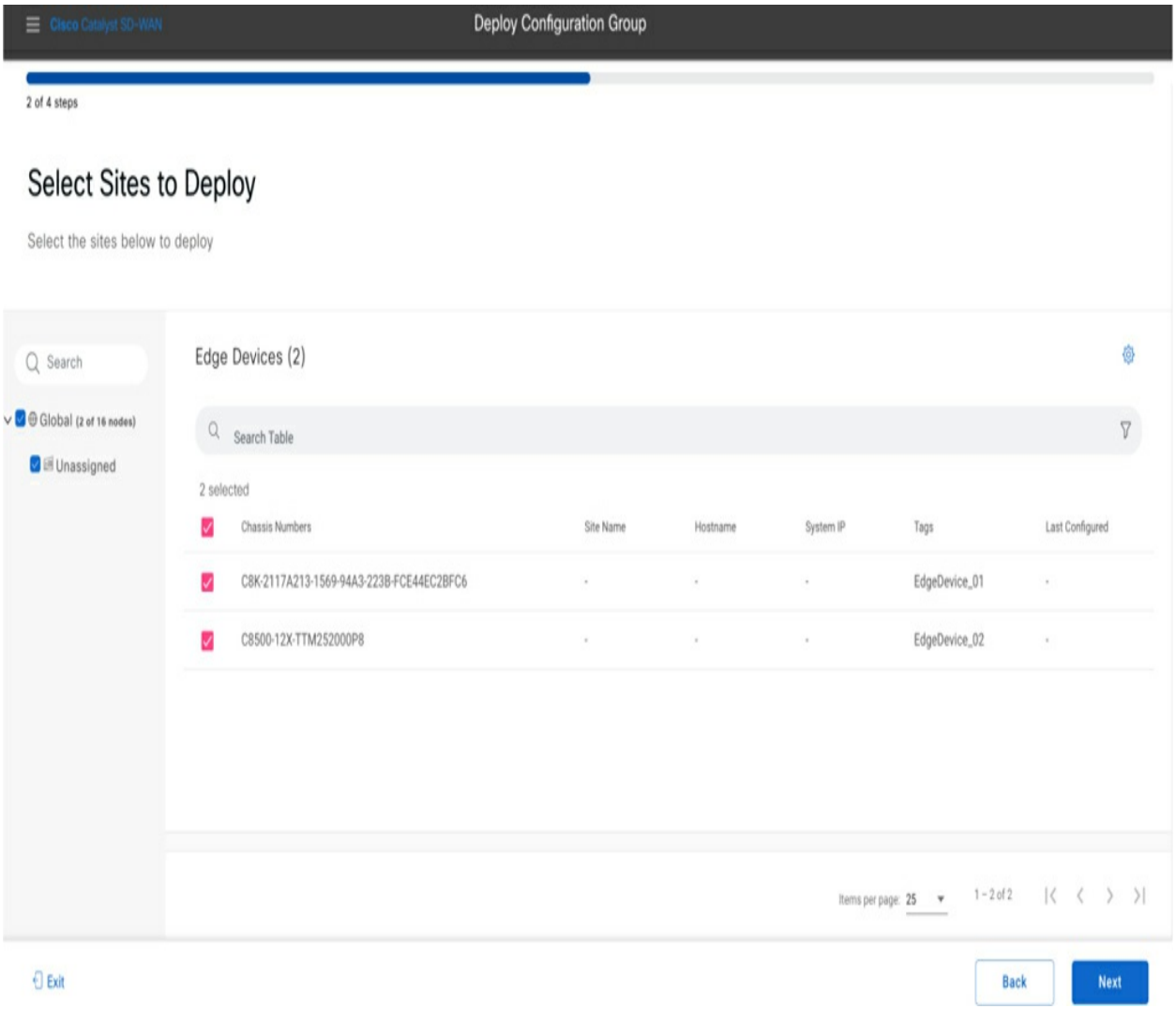


Figure 4-29 *Selecting Sites to Deploy*

Provide values for variables from nested feature parcels, either manually or via a comma-separated values file, as shown in Figure 4-30, and click **Next**. Notice that if mandatory values are missing, you won't be able to proceed to the next screen.

Technet24

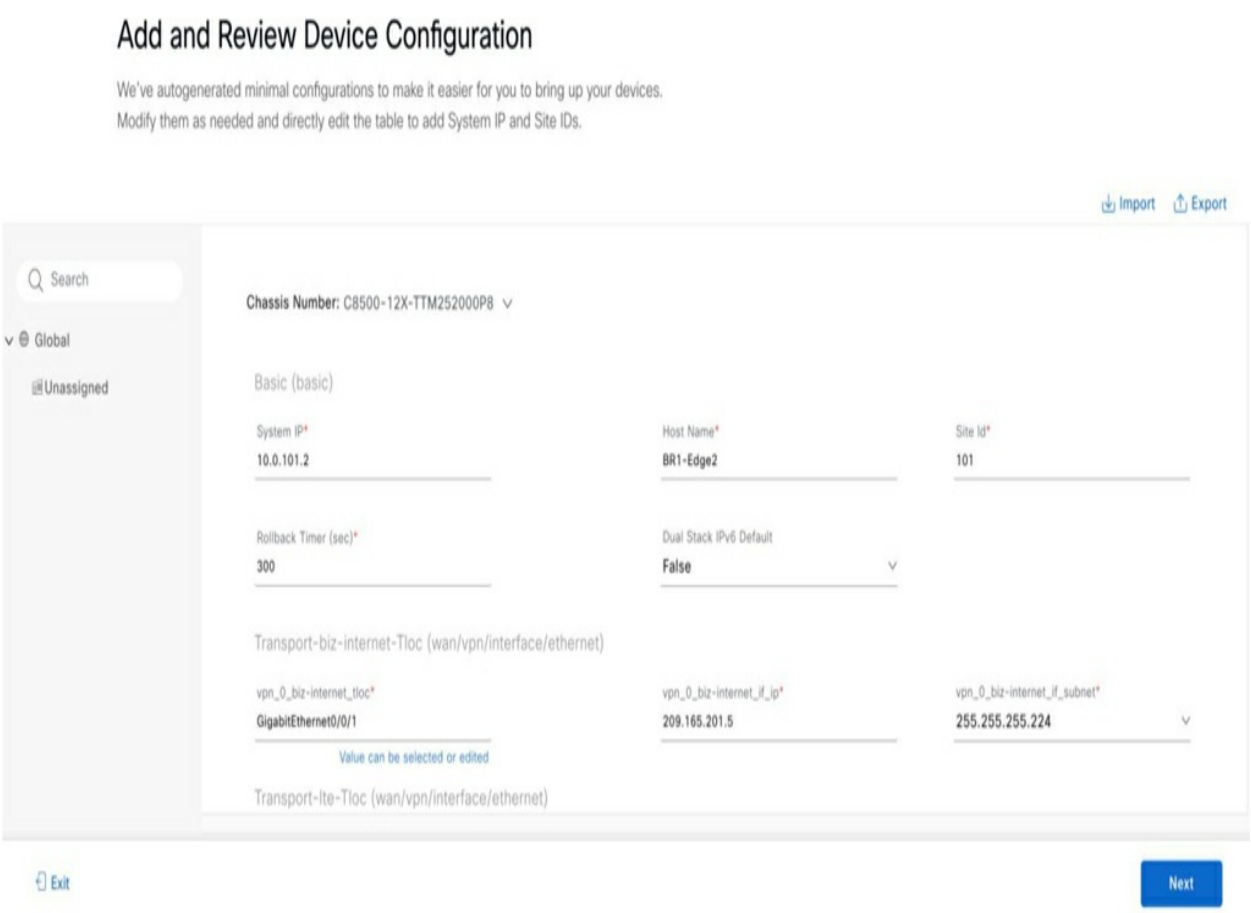


Figure 4-30 *Adding Devices and Reviewing the Device Configuration*

Examine the summary view that appears, which allows you to review the values provided during the workflow and preview the CLI configuration that you will send to devices (see Figure 4-31). If everything looks correct, click **Deploy**.

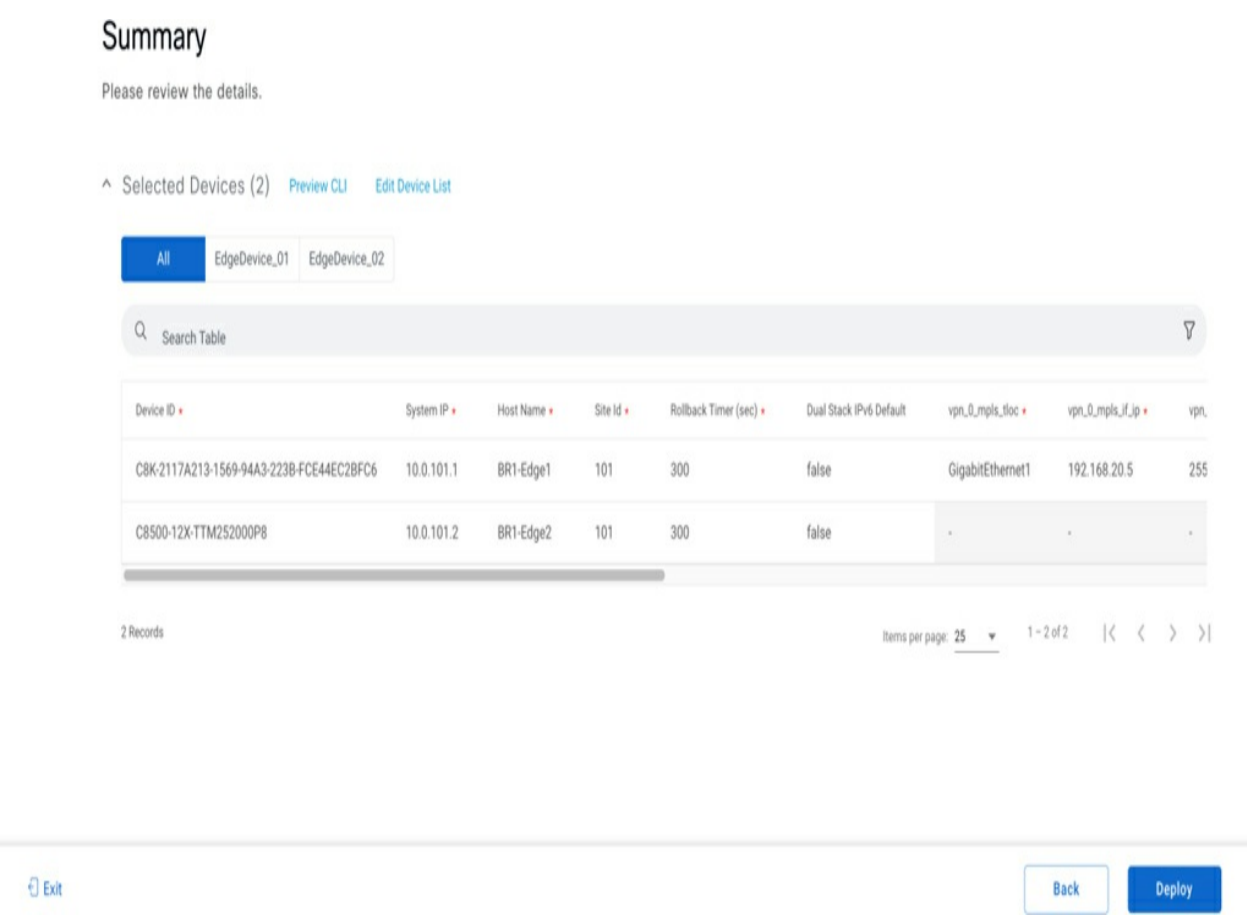


Figure 4-31 *Summary View*

Note

For both Device Templates and Configuration Groups, when a configuration change occurs, a preview CLI provides two widely used options: Inline Diff and Side by Side Diff. While both of these options highlight the differences, displaying deletions in red and additions in green, Side by Side Diff tends to be preferred because it clearly presents the old versus the new configuration the device will pull and makes it possible to revert and make required changes if needed.

Technet24

Configuration Groups offer a simplified, device-agnostic, efficient way to manage device configuration. All the relevant elements are grouped together instead of being split into many nonsequential pieces, as is the case with feature template creation and assignment to device templates. Configuration Groups also automatically create site topology to help you better visualize a site configuration's composition. You'll learn more about this in Chapter 14, "Cisco Catalyst SD-WAN Monitoring and Operations."

Be aware that feature parity between Configuration Groups and templates is progressing, and new phases are becoming available with the latest software versions. It is therefore important to test all the required functionality with Configuration Groups before you deploy them in a production environment.

As mentioned earlier, the recommended method for configuring WAN Edge routers is to leverage either Device Templates, Configuration Groups or use the SD-WAN Manager programmatic API approach. The API option streamlines the configuration process and version control compared to manually configuring individual devices.



Onboarding Devices

For a WAN Edge Device to join the SD-WAN fabric, the device first needs to establish connectivity to SD-WAN Validator (formerly vBond). SD-WAN

Validator facilitates discovery of SD-WAN Manager and SD-WAN Controller (formerly vSmart). As the WAN Edge device establishes connectivity to each of these control components, mutual authentication occurs. After the WAN Edge device has authenticated to the control components in the overlay, the device receives its full configuration from SD-WAN Manager.

There are two ways to bootstrap a device with an initial configuration so that it can reach SD-WAN Validator:

• The less preferred, and more obvious, method is to manually apply minimal configuration to the device.

• The second method is to automatically discover the network by using zero-touch provisioning (ZTP) or plug and play (PNP).

If a device is running Viptela OS, it will use ZTP. If the device is an IOS XE–based device and cloud-hosted controllers are in use, the device will use PNP, and if an on-premises ZTP server is in use, the device will use ZTP. The following sections elaborate on these processes.

A third option, generating a bootstrap file, allows an administrator to generate a basic configuration file for a WAN edge device to initially connect to the fabric or in the event of configuration loss. This option is primarily used when a network is air-gapped, for devices without Secure Unique Device Identifier (SUDI), and when employing Catalyst 8000V Edge devices.

Technet24

Manually Configuring a WAN Edge Device

To manually set up a WAN Edge device, a network administrator begins by applying a minimal configuration to the device, including IP addressing, SD-WAN Validator addressing (either DNS hostname or IP address), and system identification information. This information is used to establish initial connectivity and authentication. Example 4-1 illustrates the basic configuration required to set up a WAN Edge. The process to manually configure a device is as follows:

Configure an interface, tunnel interface, IP address, and default gateway. If DHCP is available, you can use it to assign the IP address and gateway automatically.

Configure the SD-WAN Validator IP address or hostname. If you are using a hostname, provide a DNS server address and ensure that the device has reachability from VPN 0.

Configure device identification information, including the system IP address, site ID, and organization name.

Example 4-1 *Minimal Configuration for an IOS XE–Based Device*

```
!Configuration for BR3-Edge1
Router# config-transaction
```

```
system
Router(config-system)# system-ip 10.0.103.1
Router(config-system)# site-id 103
Router (config-system)# vbond validator.ciscop
Router(config-system)# sp-organization-name Cis
Router(config-system)# organization-name Cisco-
Router(config-system)#exit
Router(config)# hostname BR3-Edge1
Router(config)# ip route 0.0.0.0 0.0.0.0 209.16
Router(config)# ip name-server 198.18.133.1
Router(config)# interface Tunnel 2
Router(config-if)# ip unnumbered  GigabitEther
Router(config-if)# tunnel source  GigabitEther
Router(config-if)# tunnel mode sdwan
Router(config)# interface  GigabitEthernet2
Router(config)# ip address 209.165.201.8 255.25
Router(config)# no shutdown
Router(config)# exit
Router(config)# sdwan
Router(config-sdwan)# interface GigabitEthernet
Router(config-interface-interface-name)# tunnel
Router(config-tunnel-interface)# color biz-inte
Router(config-tunnel-interface)# encapsulation
Router(config)# commit
BR3-Edge1# exit
```

Technet24

The tunnel interface number may seem like a random number, but it is in fact tied to the physical interface type and number through a special formula, and those two must match for the configuration to be successful. This is the formula:

Tunnel # = (Interface number without slashes) + (1000*Subinterface number, if present) + (5000000 if interface is a serial interface) + (Channel group, if present * 1000) + (10000000, if interface is TenGigabit)

For example, GigabitEthernet0/0/0 maps to Tunnel0, GigabitEthernet0/0/1.5 maps to Tunnel5001, TenGigabitEthernet0/1/0.4 maps to Tunnel10004010, and so on.

One special case is when a loopback interface sources the tunnel. In this case, the formula is as follows:

Tunnel name = 14095000 + Loopback number



Automatic Provisioning with PNP or ZTP

When provisioning devices, another option is for a network administrator to automatically bring the devices online with minimal effort and involvement. Once a device is powered on, the default configuration on the device tries to

receive an IP address via DHCP. Once the device has an IP address, it reaches out to the automatic provisioning server (hosted by Cisco) and learns about the organization's SD-WAN Validator. At this point, the process is exactly the same as the manual configuration process: The device connects and authenticates to SD-WAN Validator, learns about SD-WAN Manager and SD-WAN Controller, and then receives its configuration.

Note

The automatic provisioning servers are managed via the PNP portal at https://software.cisco.com. As devices are purchased from Cisco, their serial numbers are populated at this portal. You can also configure SD-WAN Manager to synchronize with this portal to automatically populate your organization's devices into SD-WAN Manager.

Before the automatic provisioning process can be initiated, a network administrator needs to attach a device template in SD-WAN Manager for the respective device. The device template must have the system IP address and site ID for the device populated as well. If none of this is completed, the process will not allow the administrator to proceed further in the workflow. When SD-WAN Manager sees the device for the first time, it pushes the template that is assigned to the matching serial number of the device performing ZTP or PNP.

Technet24

The process is slightly different, depending on the type of device. Cisco IOS XE devices use PNP with cloud control components or ZTP with an on-premises ZTP server, and Viptela OS–based devices use ZTP.

When a device boots up, it starts the process of receiving an IP address and DNS server via DHCP. When this succeeds, a Viptela OS device tries to resolve ztp.cisco.com, and a Cisco IOS XE device tries to resolve devicehelper.cisco.com. If it succeeds, the device connects to the provisioning server, the server verifies what organization the device belongs to, and the server redirects the device to the correct SD-WAN Validator for the organization. The provisioning server is able to verify which organization the device belongs to by checking the serial number of the device against its serial number entries database. Once the device is connected to the SD-WAN Validator, the normal process continues.

For this type of automatic provisioning method to function, two things must happen: DHCP must be available on the WAN (VPN 0)-facing interface, and the device must be able to resolve ztp.cisco.com or devicehelper.cisco.com. The process is as follows (see Figure 4-32):

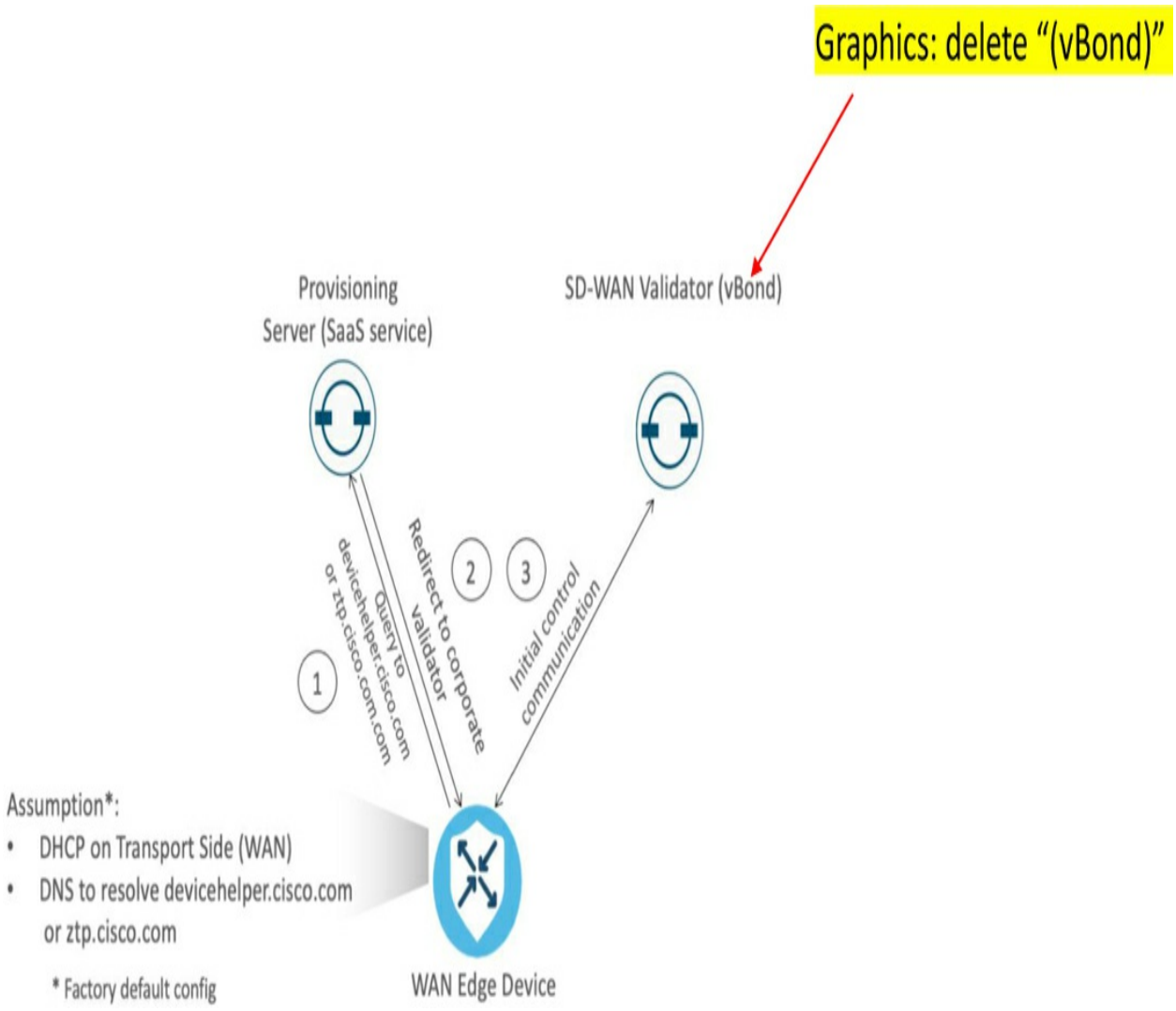


Figure 4-32 *Automatic Provisioning Workflow*

The WAN Edge router queries devicehelper.cisco.com or ztp.cisco.com. The provisioning server verifies that the device's serial number and organization exist in the serial numbers database.

If the WAN Edge router performing the automatic provisioning process exists in the database, the provisioning server responds, telling the WAN Edge router the connectivity information for the organization's SD-WAN Validator.

Technet24

The WAN Edge device connects to the corporate SD-WAN Validator and goes through the authentication process. If the process is successful, SD-WAN Validator tells the router about SD-WAN Controller and SD-WAN Manager in the overlay. At this point, the WAN Edge device pulls the necessary configuration.

The process operates almost identically for PNP and ZTP, except instead of building a DTLS tunnel to the PNP server, the device communicates to the server via HTTPS. After the PNP server validates the device, it redirects the IOS XE–based WAN Edge device to the relevant SD-WAN Validator for the organization. An IOS XE–based device faces the same requirement as a ZTP device: It must get an IP address and DNS server via DHCP and must be able to resolve devicehelper.cisco.com.

An on-premises ZTP server can be used in air-gapped networks. The PNP agent in IOS XE–based devices attempts to connect to the local server before trying to connect to a cloud-based PNP server. (This feature was introduced to IOS XE, starting with SD-WAN Version 17.3.)

Note

Starting with Version 17.7 for IOS XE devices, Cisco provides an alternative when there is no DHCP server present. After not receiving a DHCP offer, the router proceeds to detect the IP address by examining ARP packets in the network and

inferring the IP address of the interface to further progress in the PNP process. The official Cisco documentation Install and Upgrade Guides refers to this approach as “automatic IP address detection.”



Using a Bootstrap Configuration File

In some scenarios, generating a bootstrap file and loading it to a WAN Edge device’s memory can be useful. These scenarios include the lack of SUDI certificate, the use of software appliances like Catalyst 8000V, and instances where manual configuration, automatic IP address, PNP are all infeasible. There are various ways to generate a basic bootstrap file. One of the most prevalent examples is outlined in these steps.

Create a basic device template or configuration group and attach it to the target device. Then select **Configuration > Devices**, click the three dots under **Actions** for the intended device (see Figure 4-33), and choose **Generate Bootstrap Configuration**.

Technet24

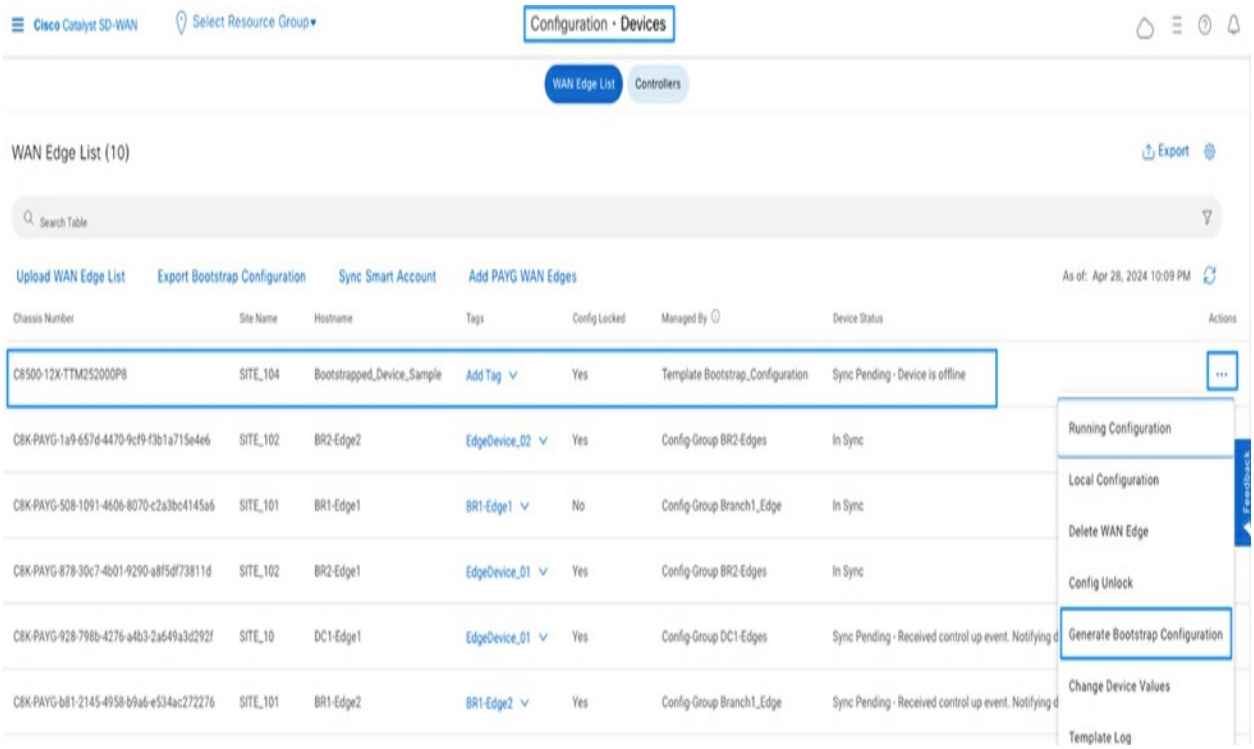


Figure 4-33 *Generating a Bootstrap Configuration*

Click **OK** and save the generated bootstrap configuration file from the popup window, as shown in Figure 4-34.

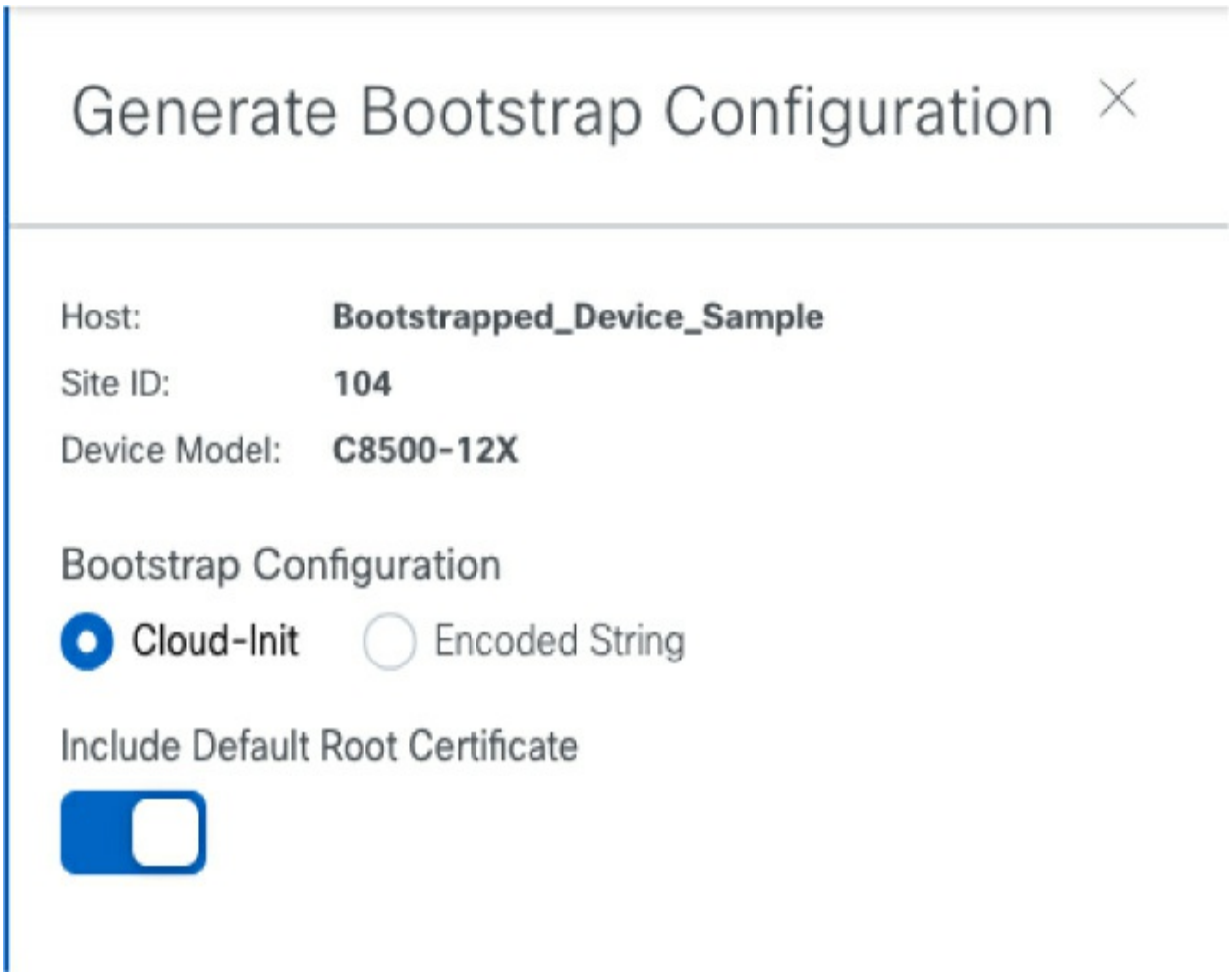


Figure 4-34 *Saving the Bootstrap Configuration*

Save the bootstrap configuration file to device memory or a USB stick or append it to the boot process of the virtual device.

Note

A bootstrap file needs to follow the naming ***ciscosdwan*.cfg** for all hardware devices except OTP-authenticated devices, such as ASR1002-X. For Cisco Catalyst 8000V edge devices and ASR1002-X, use the bootstrap filename

Technet24

***ciscosdwan_cloud_*init.cfg**. For detailed information about this process, refer to the official Cisco documentation Install and Upgrade Guides.

Depending on the network conditions, the number of steps required, or administrator preference, Cisco Catalyst SD-WAN provides flexibility by offering methods such as manual setup, automatic provisioning, or bootstrapping from a file in memory. These onboarding methods allow businesses to be adaptable.



Key Settings in Device Templates and Configuration Groups

Numerous settings can be used together depending on the business goal, particularly with Direct Internet Access, Cloud OnRamp for SaaS, and symmetric routing. Let's look at some of the frequently used settings and their use with Configuration Groups:

• **Transport interface settings:**

• **NAT:** Public transports like the Internet are commonly used not only to source IPsec tunnels but also for Direct Internet Access (DIA). To support this, public TLOC interfaces require the NAT setting to be enabled, as shown in Figure 4-35. At this writing, Version 20.12 provides three options to enable NAT: interface (the option selected in the figure), pool, and loopback.

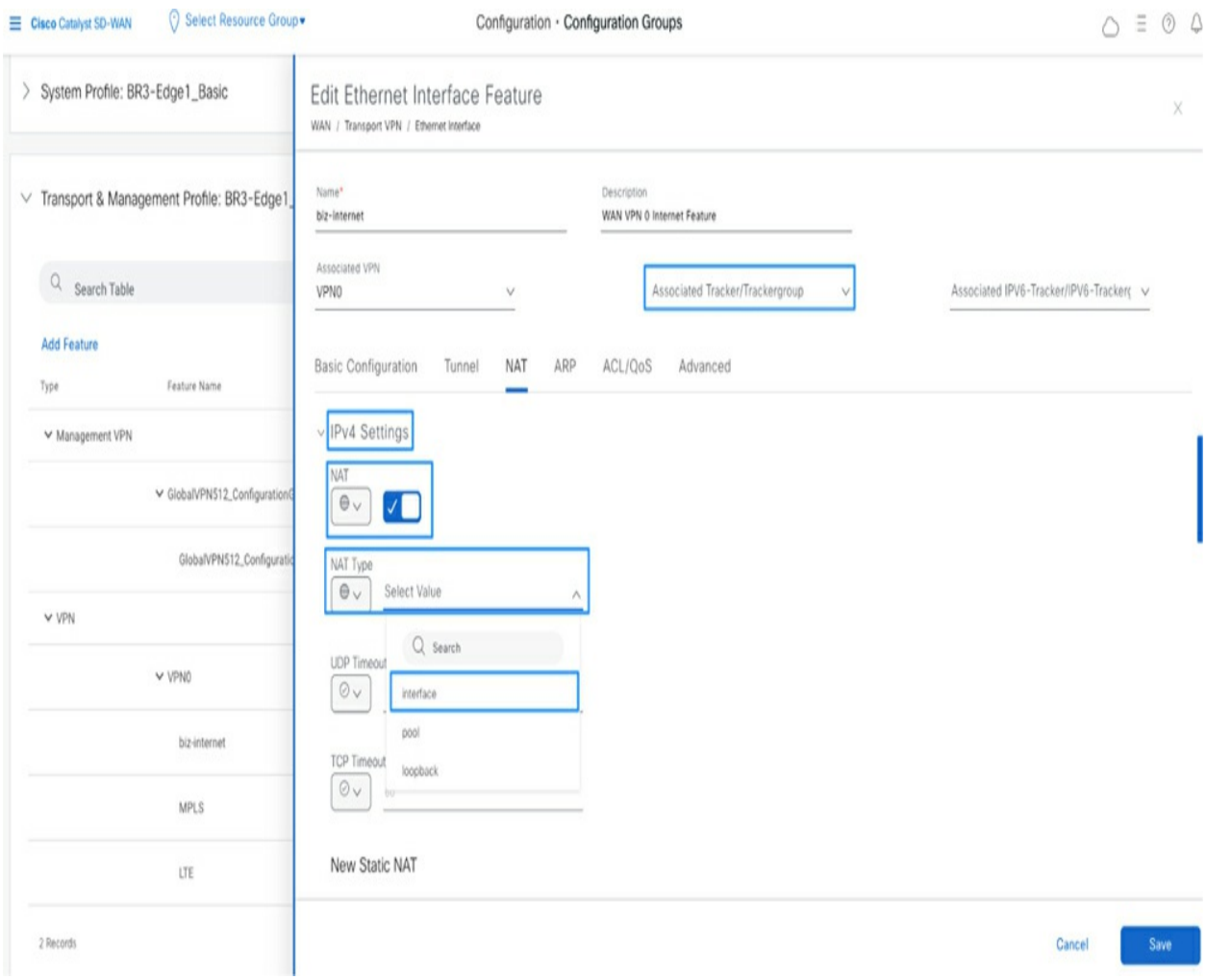


Figure 4-35 *Enabling NAT in a Public Transport Interface*

A NAT-enabled interface typically comes with a NAT DIA tracker, which is set in the Transport & Management as a Tracker Feature Parcel and then associated to the desired interface. This setting allows for probing the transport interface and detecting unavailability of the Internet or external networks. Enabling NAT on public interfaces is essential if you want to configure a NAT DIA route or selective DIA with policies. (Chapter 8, "Centralized Data Policies," discusses selective DIA.)

Technet24

• **Tunnel interface attributes:** Throughout this book, you will see private transport mappings to MPLS TLOC colors. When the **restrict** attribute is set to this TLOC, the WAN Edge device exclusively tries to establish a tunnel to any remote TLOC matching exactly that TLOC color. This means only an MPLS-to-MPLS tunnel is permitted. This attribute is commonly used within private transports to prevent the establishment of sessions to other type of transports. This setting is illustrated in Figure 4-36.

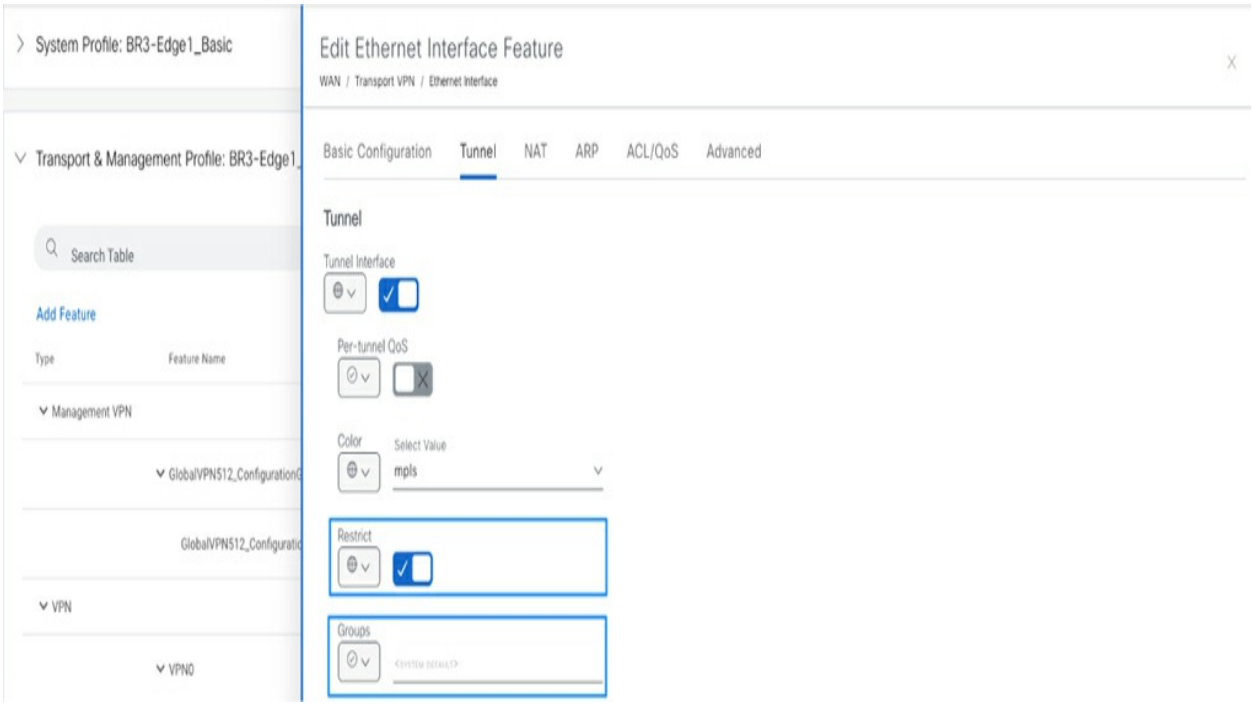


Figure 4-36 *Restrict and Groups settings under Tunnel Interface*

A similar feature is Tunnel Group ID. Although it is not configured in the use cases later described, this feature is useful when more granular tunnel creation control is needed. When the tunnel group ID matches, a tunnel is created. If there is not a tunnel group ID associated, a tunnel can be created to any tunnel group ID. When this feature is used in conjunction with the

restrict setting, both the remote color and tunnel group ID must match.

• **TLOC preference and weight:** Under normal conditions, load is equally shared across different available transports (ECMP), and 0 is the default TLOC preference value. When symmetry needs to be influenced or active/standby usage of paths needs to be implemented, TLOC preference allows you to determine the preference for traffic in the outbound and inbound directions. This is discussed in Chapter 7, "Centralized Control Policies."

The Weight attribute operates when ECMP exists. It helps to weigh the session distribution across transports. It is commonly used for sharing loads in links proportionally to the allocated bandwidth. This means an administrator can set a distribution ratio of 1 session in a 10 Mbps link for each 10 sessions used in the 100 Mbps link. These two attributes are provided in the Tunnel Interface section, as shown in Figure 4-37.

Technet24

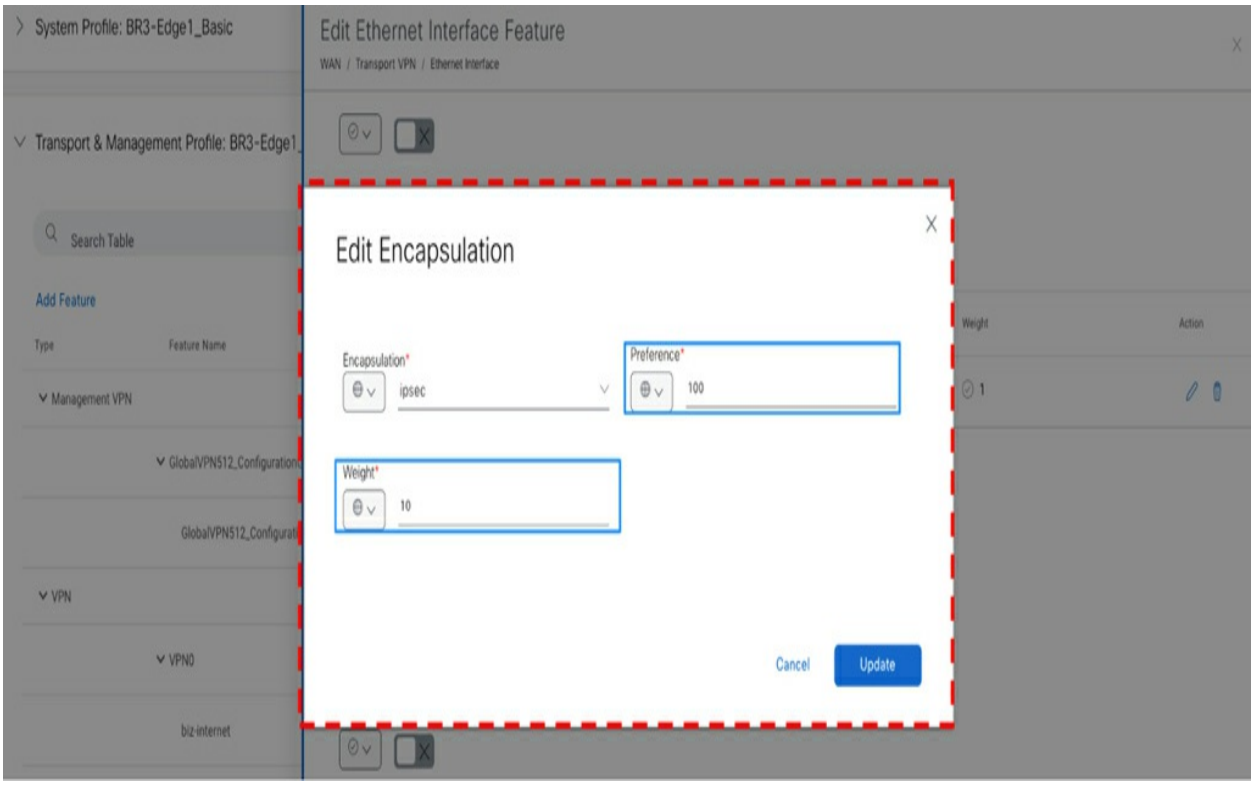


Figure 4-37 *Enabling TLOC Preference and Weight*

• **Service-side VPN settings:**

• **NAT DIA route:** In scenarios where there is a guest VPN or any data segment that needs to be connected via Direct Internet Access, employing a NAT DIA route is an option. This type of route directs all default traffic out to the Internet. It can be set with the service-side VPN settings, as shown in Figure 4-38.

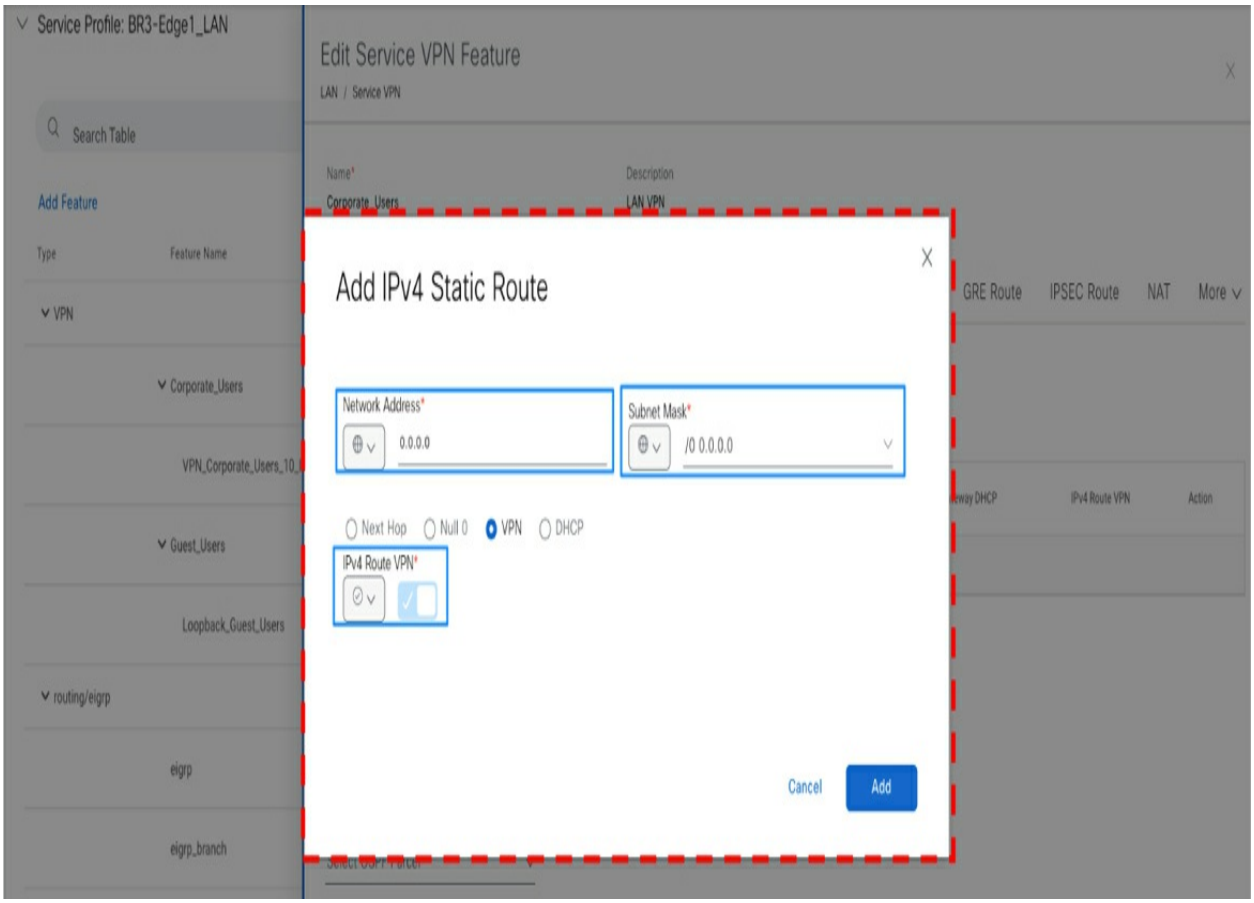


Figure 4-38 *IPv4 Route for NAT DIA*

• **Service route:** You can use a service route for traffic that is to be redirected to a Secure Internet Gateway tunnel (see Figure 4-39P.

Technet24

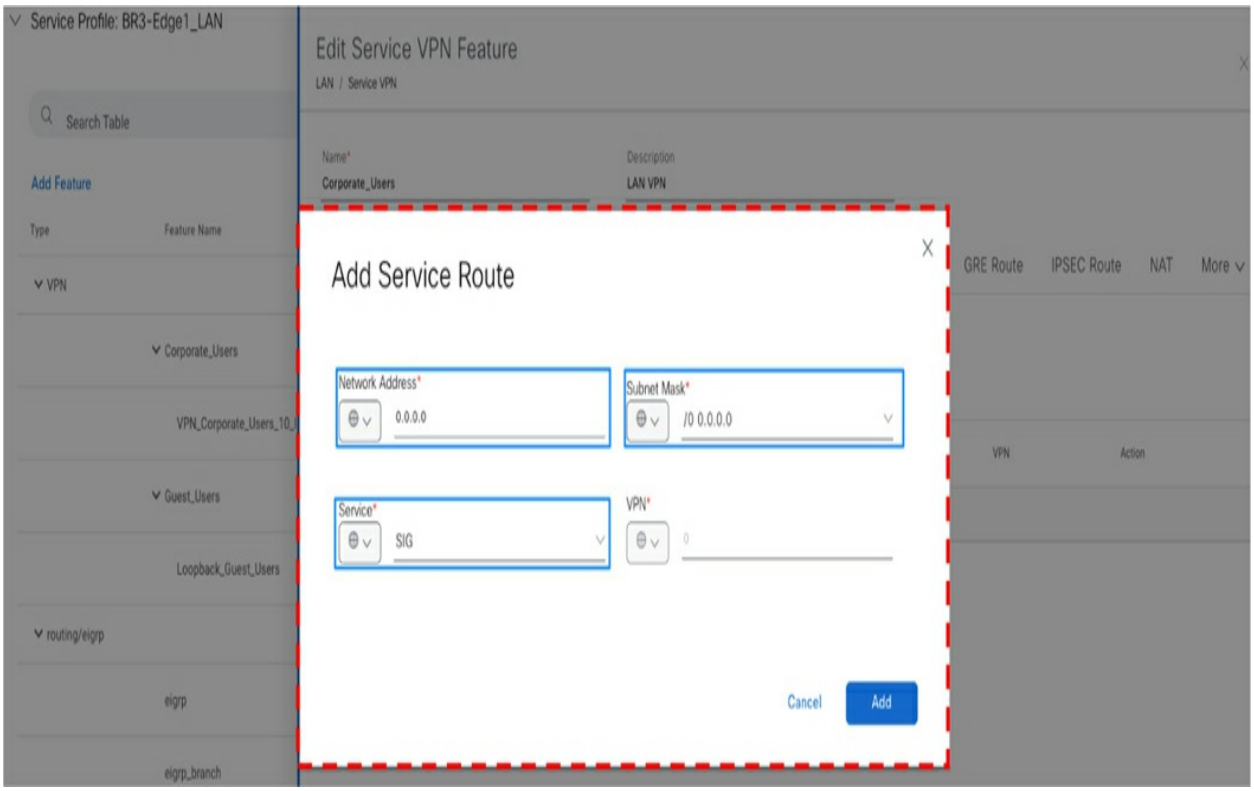


Figure 4-39 *Secure Internet Gateway Service Route*

When you need selective traffic redirection to the Internet or a Secure Internet Gateway provider, you want to use data policies, as discussed in Chapter 8.

As discussed earlier, numerous settings offer administrators options to fulfill various use cases, including ACLs, QoS, or route leaking of services between a global VPN and a service VPN. For more comprehensive information about available options, refer to the Cisco Catalyst SD-WAN Systems and Interfaces Configuration Guide for the software version you are using.

Summary

With Cisco Catalyst SD-WAN, network settings are handled with ease via a powerful configuration engine and templates or configuration groups. Feature templates can be used to achieve modular configurations that can be reused across various platforms. Feature templates are used to form a device template. Configuration groups present a hierarchical approach that offers more structure and addresses scalability, reusability, and ease of configuration. By using variables within these two constructs, a network administrator can support a large range of configuration requirements.

Provisioning of devices can occur with one of three methods: either manually, with an automatic process such as PNP or ZTP, or through the use of a bootstrap file. With the manual method, the amount of configuration is very small. The device must have a system IP address, site ID, organization name, and IP address information, and it must have a tunnel interface defined. When a device authenticates to the organization's SD-WAN Validator, it discovers the rest of the controller elements, and SD-WAN Manager can provide configuration to the device. The second option involves using an automatic onboarding process, via either ZTP or PNP, depending on whether it's a Viptela OS–based product or a Cisco IOS XE–based product. Once the device has gone through the automatic method and determined the SD-WAN Validator to use, the process works just the same as the manual method. Automatic onboarding allows a network administrator to deploy a large number of devices quickly, even if they have different configuration options.

Technet24

Review All Key Topics

Review the most important topics in the chapter, noted with the Key Topic icon in the outer margin of the page. Table 4-1 lists these key topics and the page number on which each is found.



Table 4-1 *Key Topics*

Key Topic Element	Description	Page
Paragraphs	Device templates	
Paragraph	Feature templates	
Section	Device Onboarding methods	
Section	Automatic provisioning with PNP or ZTP	
Section	Developing Configuration Groups, Feature Profiles and Feature Parcels	
Section	Use of Bootstrap configuration file	
Section	Key settings such as NAT Enabled interfaces, Tracker, Restrict, Tunnel Groups, NAT DIA Route, Service Route	

Key Terms

Define the following key terms from this chapter and check your answers in the glossary:

Device Template

Technet24

Feature Template

Configuration Group

Feature Profile

Feature Parcel

SUDI

Chapter Review Questions

1. What methods can be used to construct device templates? (Choose three.)

a. Using the CLI

b. Using feature templates

c. Directly using the device

d. Using a programmatic approach (APIs)

2. What device value types can be used with feature templates? (Choose three.)

a. Global

b. Default

c. Automatic

d. Imported

e. Device specific (variables)

3. True or false: A single device template supports multiple different device types.

a. True

b. False

4. True or false: CLI templates can be used modularly and achieve the same flexibility as feature templates.

a. False

b. True

5. Which automatic provisioning method uses HTTPS for communication?

a. Plug and play

b. Zero touch provisioning

c. NAT traversal

Technet24

6. What must a device have for automatic provisioning to be successful? (Choose three.)

a. IP address and DNS server via DHCP

b. Ability to resolve a ZTP/PNP domain name

c. Connectivity to a ZTP or PNP server

d. IPsec tunnel

e. Connectivity to the data center

7. True or false: A single configuration group supports multiple different device types.

a. True

b. False

8. True or false: Devices can be associated with both configuration groups and templates at the same time.

a. True

b. False

9. True or false: In order to use Direct Internet Access over public interfaces,

NAT needs to be enabled on them.

a. True

b. False

References

“Cisco Catalyst SD-WAN Command Line Reference,” https://www.cisco.com/c/en/us/td/docs/routers/sdwan/command/sdwan-cr-book.html.

“Cisco Catalyst SD-WAN Control Components Certificates and Authorized Serial Number File Prescriptive Deployment Guide,” https://www.cisco.com/c/en/us/td/docs/solutions/CVD/SDWAN/cisco-sdwan-controller-cert-deploy-guide.html, December 2023.

“Configuration Groups,” https://www.cisco.com/c/en/us/td/docs/routers/sdwan/configuration/config-groups/configuration-group-guide/introduction.html.

Technet24

Chapter 5. Cisco Catalyst SD-WAN Design and Migration

This chapter covers the following topics:

• **Cisco Catalyst SD-WAN Design:** This section outlines the methodology behind SD-WAN design across an enterprise.

• **Cisco Catalyst SD-WAN Control Components Design:** This section details the design options for Cisco Catalyst SD-WAN Control Components.

• **Cisco Catalyst SD-WAN Implementation Preparation:** This section covers the recommended steps to prepare for implementing Cisco Catalyst SD-WAN in an enterprise.

• **Cisco SD-WAN Transport Connectivity:** This section reviews various options for how WAN Edge routers can use available underlay transports connections.

• **Cisco Catalyst SD-WAN Data Center Design:** This section provides details on data center design options and migration techniques.

• **Cisco Catalyst SD-WAN Branch Design:** This section explores the details of branch design options and migration techniques.

• **Integrating Cisco SD-WAN with Existing Networks:** This section

discusses optimal connectivity methods between migrated and non-migrated sites.

For many organizations, the chance to deploy Cisco Catalyst SD-WAN (Cisco SD-WAN) in a greenfield environment is rare. Instead, most opportunities to implement it occur in brownfield environments, offering a unique chance to re-architect the entire wide-area network.

The WAN is such a critical and sensitive component of a typical organization’s business that approvals for lengthy outages necessitated by pervasive design changes are a rarity. Most organizations therefore see many years of incremental changes done by a multitude of individuals (with different skillsets and design methodologies) along with temporary fixes or adjustments that have never been reevaluated. Coupled with inconsistent configurations and connectivity models, this all results in complexities that, oftentimes, not a single person on staff may fully comprehend.

Designing your SD-WAN network is a golden opportunity to really dig deep into your WAN. You can learn how it operates, understand all the complexities and caveats that lie within, and propose a scalable, high-performing, easy-to-manage next-generation solution. It is crucial that the individuals leading the Cisco SD-WAN design and deployment dissect the existing WAN architecture in order to gain a solid understanding of routing, topology, high availability, failover, and traffic flow patterns as well areas where the current WAN architecture may not meet business requirements.

Technet24

The goal is to discover all the potential problems that may arise during implementation. With this information in hand, you can utilize the incredibly comprehensive tool sets provided by Cisco SD-WAN to effectively solve these problems intelligently and efficiently. During this process, the ultimate goal should be to remove unnecessary complexity, either through the evaluation of why that complexity existed initially and whether it is still required or through replacing it with the intelligent application-aware and flexible fabric that Cisco SD-WAN offers.

This section presents design recommendations garnered through years of Cisco SD-WAN implementation experience that will allow an organization to gracefully migrate to this new architecture while ensuring that the existing WAN is not disrupted. Ideally, the legacy WAN and the Cisco SD-WAN should be able to operate in parallel. An example of such a recommendation might be to selectively place Cisco SD-WAN hubs across your enterprise in order to act as transit points between migrated and non-migrated sites. Another recommendation might be to provide high availability at branch sites through the use of Virtual Router Redundancy Protocol (VRRP) between an existing router and the new SD-WAN router.

There are many ways to implement Cisco SD-WAN, and some may be better than others. In the end, however, everything always comes back to the fundamental routing and traffic engineering concepts we know and love today. Rest assured that Cisco SD-WAN provides comprehensive routing and traffic engineering tool sets, including best path manipulation options,

advanced routing protocol features, filtering and tagging, and an overlay management protocol that delivers unprecedented traffic engineering flexibility.

Cisco SD-WAN Design Methodology



Crafting a well-thought-out design for the SD-WAN fabric is a crucial step in the journey to Cisco SD-WAN enablement. It is a key to ensuring success across all stages of implementation, including setting up SD-WAN Control Components, deploying data centers and branches, and configuring policies.

Cisco recommends taking a top-down approach to SD-WAN designs to ensure that the chosen platforms, use cases, and features align with the organization's implementation goals and business objectives. Implementations that follow SD-WAN design methodology typically include these considerations:

• **SD-WAN objectives and requirements:** Determine the reason for implementing SD-WAN network, along with objectives and requirements for the network.

• **Network audit:** Identify current applications in the network, their latency and loss tolerances, and traffic patterns. Review current hardware, software, and transports, along with their utilization. Evaluate network performance to

Technet24

identify any problem areas.

• **WAN and site standards:** Develop standards for physical and logical topology (site types), dictating how the WAN Edge routers connect to the LAN and WAN infrastructure. Consider different site types to accommodate remote sites with varying business requirements, numbers of users, or available transports. Ensure that site standards account for future SD-WAN use cases and features that may impact traffic load, flow patterns, and other factors influencing decisions on bandwidth, redundancy, and WAN Edge platform selection.

• **High-level design:** Make decisions regarding key elements of the SD-WAN deployment, such as budget allocation, implementation scope, and orders for circuits and hardware. High-level design includes the information needed to determine the platform choices, such as required features, circuits, bandwidth, and number of data tunnels per device for each site in the design.

• **Low-level design:** Develop and validate specifics about configurations, policies, and other aspects of the implementation. This work can be done in parallel with the high-level design.

Subsequent sections of this chapter explore specific aspects of the SD-WAN design methodology in greater depth.

Cisco SD-WAN Objectives

Designing an SD-WAN network from scratch is a complex assignment. It is recommended to approach it in a structured manner, such as by taking a top-down design approach, analyzing business, functional, and technical (non-functional) requirements first before deciding on technologies and features.

Table 5-1 provides few examples of common business objectives that can be addressed by a Cisco SD-WAN deployment.

Table 5-1 *Business Objectives Addressed by Cisco SD-WAN*

Technet24

Objective	How Cisco SD-WAN Addresses Objective
Availability and performance	Cisco SD-WAN allows businesses to use all available network transports at full capacity, using automated path control to prioritize business-critical applications and improve performance.
Optimized cloud architecture	Cisco SD- WAN optimizes networks to connect to cloud-based applications easily and efficiently. It provides direct secure, high- performance connections to the cloud, avoiding backhauling traffic to data centers. With Cisco SD-WAN, the user experience with IaaS/Colo/SaaS- based applications is greatly improved.
Savings	Cisco SD-WAN makes it possible to replace expensive technologies (such as MPLS) with more cost-effective connectivity options (such as broadband and LTE). It reduces networking costs while easing demands on IT, thus increasing productivity and profitability.
Efficiency	Cisco SD-WAN provides a simplified and centralized configuration, policy management, network monitoring, and troubleshooting platform, making it easy to deploy and monitor individual sites and applications.
Security	Cisco SD- WAN delivers an integrated security platform with built-in user identity–based control, data encryption, and firewall support, using network intelligence to detect and respond to cyberattacks in real time.
Agility	Cisco SD-WAN enables zero-touch provisioning, reducing deployment time across the network.

Cisco SD-WAN Use Cases

As you analyze your specific requirements, you will likely find that the Cisco SD-WAN functionality that suits your needs aligns with one or more standard categories. There are four main use cases that Cisco SD-WAN is designed to efficiently address:

• Secure automated WAN

• Application performance optimization

• Secure direct Internet access

• Multicloud connectivity

These categories are not mutually exclusive, and a Cisco SD-WAN deployment may incorporate any combination of these use cases, depending on the specific needs and objectives of the network. Let’s briefly review each of these use cases.

Use Case 1: Secure Automated WAN

The goal of the secure automated WAN use case is to establish secure connectivity between all network locations, including branches, data centers, colocations, and both public and private clouds, over a transport-independent network. This is the most typical use case for Cisco SD-WAN and forms the foundation for other use cases.

Technet24

A secure automation WAN includes the following functionalities:

• **Transport flexibility and bandwidth augmentation:** Cisco SD-WAN supports various transport types, such as MPLS, Internet, 4G/5G LTE, satellite links, and more. Organizations can select the most suitable transport options based on cost, performance, and availability. Cisco SD-WAN can efficiently use available transports, whether by distributing traffic across all available WAN circuits in an active/active manner to increase overall WAN bandwidth or by intelligently offloading traffic from expensive circuits like MPLS to more cost-efficient broadband connections, all while maintaining equivalent availability and performance.

• **VPN segmentation:** Cisco SD-WAN creates routing table isolation, ensuring that a user in one VPN cannot transmit data to another VPN unless this capability is explicitly configured.

• **Zero-touch provisioning:** Cisco SD-WAN enables you to remotely provision a router anywhere in the WAN by simply connecting it to the transport network and powering it on.

• **Centralized management:** SD-WAN Manager offers a single pane of glass for centralized management of SD-WAN operations. It provides operational simplicity and streamlines deployment by using global policies and templates, resulting in reduced change control and deployment times.

Use Case 2: Application Performance Optimization

Achieving the best application experience is critical for user productivity, especially when a network experiences issues like packet loss, congested WAN links, and suboptimal WAN routing. Cisco SD-WAN excels at optimizing application performance and mitigating the impact of WAN latency and forwarding delays through the following capabilities:

• **Application-Aware Routing (AAR):** Transport performance is measured in real time and checked against customized SLA metrics. The application traffic is directed to WAN links that are in compliance with the SLAs for that application but can be directed to other paths during periods of performance degradation.

• **Quality of Service (QoS):** QoS uses classification, scheduling, queueing, shaping, and policing of traffic on the WAN router interfaces to minimize delay, jitter, and packet loss with critical application flows.

• **Forward Error Correction (FEC) and packet duplication:** These features are used to mitigate packet loss. With FEC, the transmitting WAN Edge router inserts a parity packet for every four data packets, and the receiving WAN Edge can reconstruct a lost packet based on the parity value. With packet duplication, the transmitting WAN Edge router replicates all packets for selected critical applications over two tunnels, and the receiving router reconstructs critical application flows, discarding duplicates.

• **TCP optimization and session persistence:** These features can address

Technet24

high latency and poor throughput for long-haul or high-latency links such as satellite.

• **Data Redundancy Elimination (DRE):** This feature is a type of TCP optimization that uses compression technology to remove redundant information, thus reducing the size of the transmitted data across the WAN.

Chapter 8, "Centralized Data Policies," and Chapter 9, "Application-Aware Routing Policies," cover application performance optimization use cases in more detail.

Use Case 3: Secure Direct Internet Access

In a traditional WAN, Internet traffic from a branch site is backhauled to centralized locations such as data centers, where it can be scrubbed by a security stack. Today, as more companies embrace cloud services for their applications, and the prevalence of Internet-based applications continues to grow, this approach leads to increased Internet traffic. This surge places additional strain on the security and network devices at central sites, increased bandwidth utilization, and suboptimal latency, negatively impacting application performance.

Direct Internet Access (DIA) and Direct Cloud Access (DCA) present a solution for these challenges by allowing Internet-bound user traffic—either all traffic or a subset of it—to exit locally from the remote site, as shown on Figure 5-1.

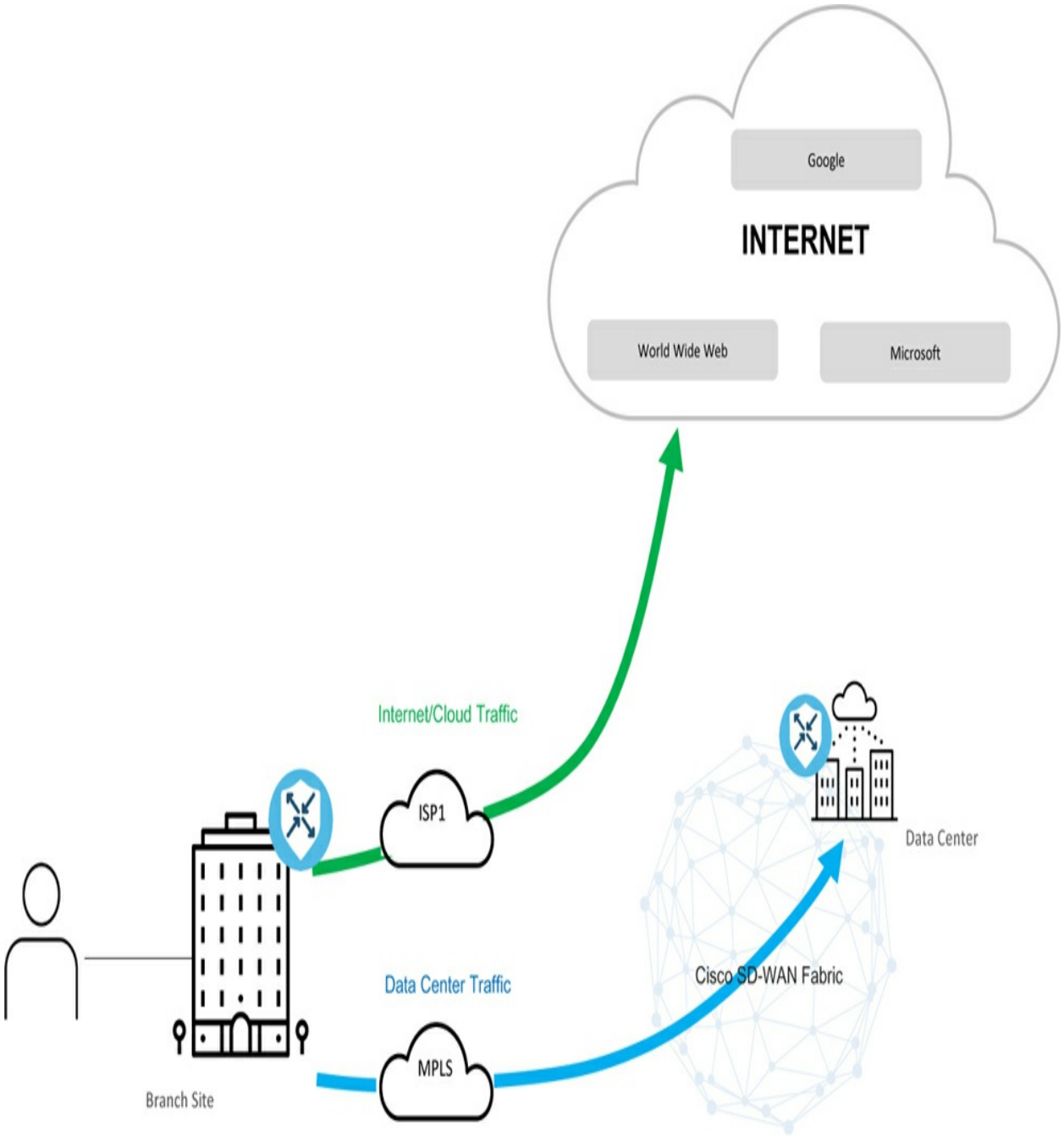


Figure 5-1 *Direct Internet Access/Direct Cloud Access*

Note

Keep in mind that merely using Internet transport as an underlay to reach a data center through an SD-WAN tunnel

Technet24

for centralized Internet access does not constitute direct Internet access. The key distinction is a localized diversion of the Internet-bound traffic to a local Internet link.

While offering efficient Internet access, DIA introduces security challenges, such as safeguarding remote site traffic against Internet threats. Cisco SD-WAN addresses this with the embedded security features on WAN Edge devices or by leveraging a secure access service edge (SASE) model with cloud-based security providers—or a combination of both. Cloud providers include Cisco Umbrella for DNS resolution, Cisco Secure Access Secure Service Edge (SSE), and other third-party secure Internet gateway (SIG) provides, like Zscaler and Netskope. The SASE model enhances security by consolidating various networking and security functions into a single integrated cloud service, ensuring secure application access for users regardless of their location.

The embedded security features in WAN Edge routers include an enterprise firewall with application awareness, an intrusion detection system (IDS)/intrusion prevention system (IPS), DNS security, URL Filtering, SSL proxy, and advanced malware protection.

The Cisco Umbrella cloud-based service provides a suite of security features, including a secure web gateway, DNS-layer security, cloud-delivered firewall, cloud access security broker functionality, and threat intelligence. In addition, the Cisco SASE model includes Cisco Duo, which offers two-factor

authentication and endpoint security, and Cisco ThousandEyes, which offers Internet and cloud visibility to ensure exceptional user application experiences.

With Cisco Secure Access, Cisco Umbrella, and Zscaler, Cisco SD-WAN can leverage the SIG APIs to automate tunnel provisioning. This capability streamlines deployment for users, enabling quick and efficient setup with minimal configuration.

Chapter 8, "Centralized Data Policies," and Chapter 11, "Cisco Catalyst SD-WAN Security," cover secure direct Internet access use cases in more detail.

Use Case 4: Multicloud Connectivity

Organizations increasingly rely on a variety of cloud services, each with unique features and advantages. Multicloud connectivity use cases focus on ensuring that users consistently experience low latency and high performance when accessing cloud applications, regardless of the specific cloud services involved.

The following use cases are associated with this category:

• **Infrastructure as a service (IaaS):** IaaS is a public cloud offering (such as AWS, Azure, or Google Cloud) that provides on-demand access to network, compute, and storage resources over the Internet. Traditionally, branches lacked direct access to cloud IaaS resources and required connectivity

Technet24

through a data center or colocation site, as highlighted in the previous section.

Cisco Cloud OnRamp for Multicloud simplifies connectivity to workloads in the public cloud. It automatically deploys WAN Edge router instances in the cloud and integrates them into the SD-WAN fabric, establishing data plane connectivity to the routers in data centers or branches. This extension of full SD-WAN capabilities into the cloud ensures a common policy framework across the entire SD-WAN fabric and cloud. Consequently, Cisco Cloud OnRamp for Multicloud enhances application performance by eliminating the need for SD-WAN traffic to traverse the data center when accessing applications hosted in the public cloud.

Figure 5-2 illustrates this use case:

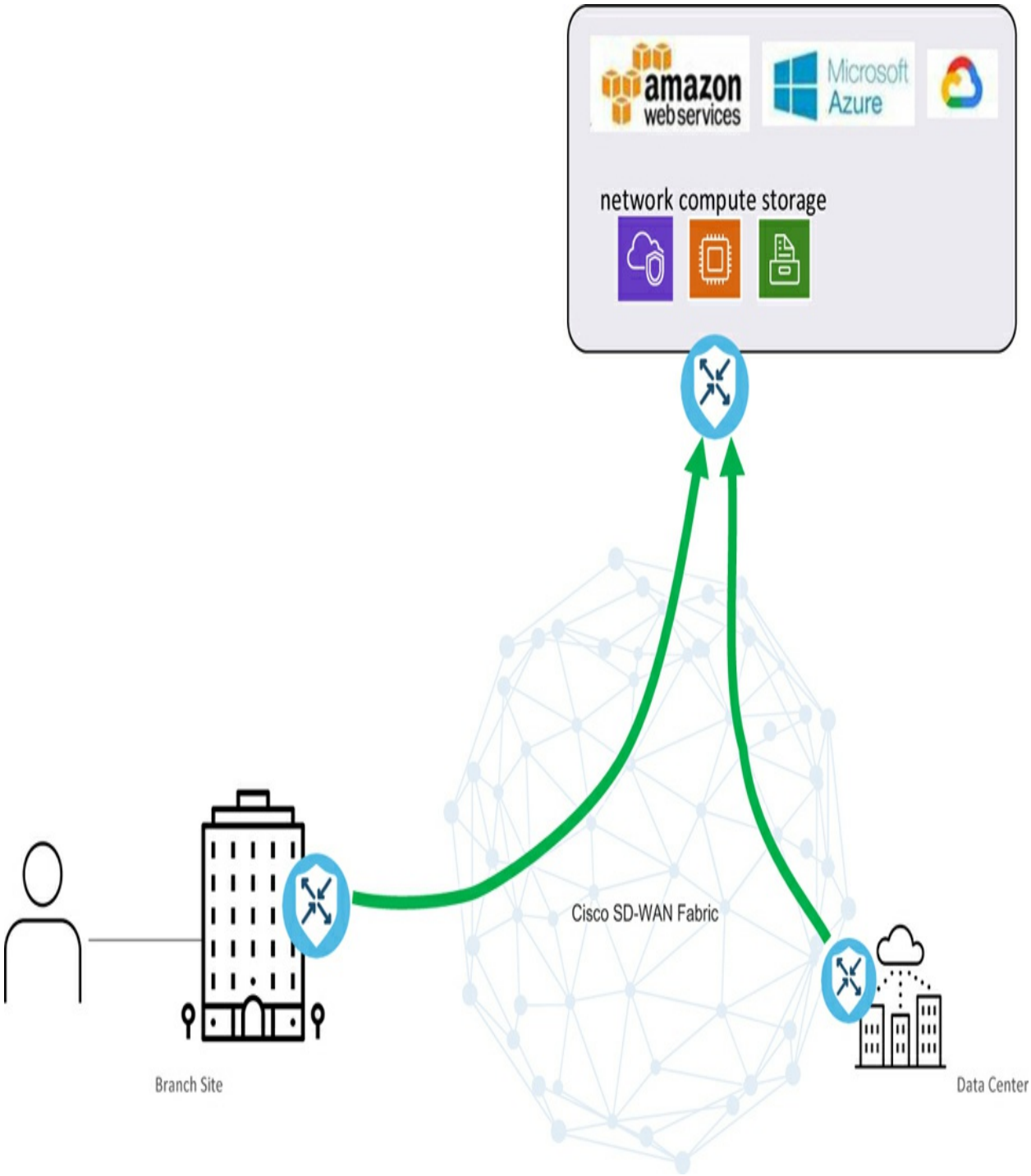


Figure 5-2 *Cisco Cloud OnRamp for Multicloud Use Case*

• **Software as a service (SaaS):** Traditionally, branches accessed cloud SaaS applications (Office 365, Salesforce, Box, and so on) through centralized data

Technet24

centers, leading to increased latency and an unpredictable user experience. SD-WAN introduced additional network paths for accessing SaaS applications, such as direct Internet access and connections through regional gateways. However, optimal path selection for accessing SaaS applications becomes challenging when the network lacks visibility into the performance of these applications from remote sites.

Cloud OnRamp for SaaS facilitates easily configurable access to SaaS applications, offering the flexibility of direct access via the local Internet or through designated gateway locations. As shown in Figure 5-3, the router continuously probes, measures, and monitors the performance of each path to every SaaS application and dynamically selects the best-performing path based on loss and delay. In the event of any impairment, SaaS traffic is intelligently shifted to the updated optimal path, ensuring a seamless and optimal user experience.

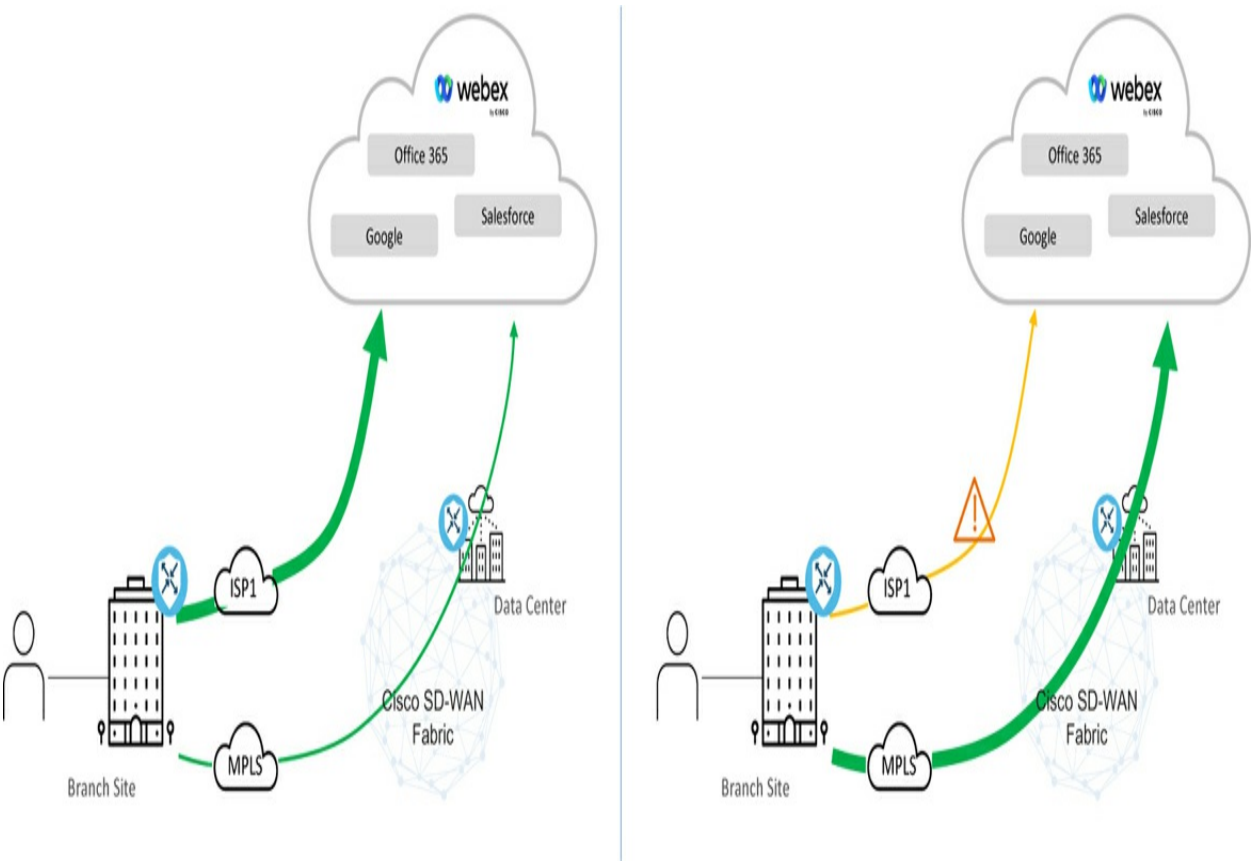


Figure 5-3 *Cisco Cloud OnRamp for SaaS Use Case*

Expanding on the basic benefits of Cloud OnRamp for SaaS, additional features have been introduced to enhance integration with Office 365. These features provide users with more insightful metrics, increased control over traffic flow for individual Office 365 applications, and automatic remediation of suboptimal performance, leveraging Microsoft telemetry metrics for a more refined and efficient user experience.

• **Software-defined cloud interconnect (SDCI):** Traditionally, sites connecting to cloud applications often relied on using transports like Internet and MPLS, which could be unreliable, insecure, or both. SDCI addresses these concerns by facilitating site-to-site and site-to-cloud connections

Technet24

through geographically dispersed points-of-presence (POPs) located in colocation facilities. These POPs interconnect with multiple ISPs and cloud providers at speeds of 100 Gbps and higher.

The SDCI approach offers quick and flexible software-defined onboarding that requires no additional hardware investment from the customer. This is accomplished by automatically deploying a virtual Catalyst 8000V WAN Edge router into the SDCI provider and integrating it into the SD-WAN fabric.

The customer can then use its preferred transports to reach the nearest SDCI provider's POP and rely on Cisco SD-WAN for optimized traffic flows. From there, user traffic moves onto the backbone of the SDCI provider, and it is then forwarded to the appropriate cloud providers. This technique provides reliable and dedicated bandwidth that is secure and cost-effective.

At this writing, Cisco SD-WAN supports Equinix and Megaport SDCI providers.

Chapter 12, "Cisco Catalyst SD-WAN Cloud OnRamp," covers multicloud connectivity use cases in more detail.

Cisco SD-WAN High-Level Design Considerations

During the high-level design phase, critical decisions are made regarding key elements of the SD-WAN deployment. The high-level design guides budget

allocation and orders for hardware, software, licenses, control components, WAN transport circuits, and so on. Completing the high-level design early is crucial as some of these items may require extended lead times, especially when site construction is required to install new transports or hardware delivery is delayed due to manufacturing backlogs.

The high-level design should address the following topics:

• **Planned use cases and features:** Review the Cisco SD-WAN use cases and features, such as application visibility, VPN segmentation, AAR, QoS, FEC, cloud access, on-premises security, and multicast, and decide which of them will be implemented.

• **Cisco SD-WAN software version:** The code version used in the deployment can influence the design based on the features supported as well as the control plane scale. When choosing software for SD-WAN Control Components and WAN Edge routers, it is generally recommended to use versions that are marked with a gold star (based on software quality, stability, and longevity) at the Cisco Software Download site. However, you may require newer software that has new functionality or supports newer hardware, in which case it is recommended to use extended maintenance releases (for example, 20.9, 20.12, 20.15) wherever possible. You can find more details about current software recommendations from Cisco in the "Cisco Recommended SD-WAN Software Versions for Controllers and WAN Edge Routers" document at the Cisco website.

Technet24

• **Transport strategy:** Evaluate the number of carriers and transport types for your SD-WAN design. Determine whether these transports will be interconnected or independent and determine their availability at data centers and in branches, bandwidth capacities, and whether there is a preferred transport. Keep in mind that while Cisco SD-WAN features such as AAR and FEC can mitigate WAN impairments, they are not a substitute for reliable transports provisioned with sufficient bandwidth.

• **Transport color scheme:** Determine what colors are assigned to transports in the SD-WAN overlay. Once the scheme is developed, you can use tools like color restriction, tunnel groups, and centralized control policies to manage the number of data tunnels between routers. Recall that WAN Edge routers by default attempt to build data tunnels from every local TLOC to every remote TLOC in the network, which increases the number of data tunnels in the SD-WAN network and limits the network's scalability.

• **VPN segmentation:** Identify segmentation requirements and establish the number of service VPNs in the SD-WAN network. Assess whether certain VPNs need to enable communication with other VPNs within a single site or across multiple sites, such as connecting to a shared VPN in a data center.

• **Overlay topology choices:** Based on the network audit and identified traffic flow patterns, decide whether remote sites should only form data tunnels to data centers in a hub-and-spoke topology, or if there's a need for full-mesh connectivity between sites for specific applications, such as voice.

• **Deployment scale**: Evaluate the number of data center and branch sites, as well as the total count of routers. These factors significantly impact the network's scalability. Future-proof your design by planning for potential growth, ensuring that the network can adapt and scale as needed. Determine whether your network's scale requires a Multi-Region Fabric architecture (as discussed shortly).

• **Deployment of SD-WAN Control Components:** Determine whether SD-WAN Control Components will be hosted in the cloud or on premises. Calculate the total number of control connections from WAN Edge routers based on the deployment scale and the color scheme. Use this information to establish the quantity of SD-WAN Control Components needed to effectively support the current and future scale of the network, considering potential growth.

For an on-premises deployment, identify the number of physical servers required to host of SD-WAN Control Components, along with their characteristics (CPU, memory, interfaces, disk space, and so on). Plan for the worst case and future growth, as adding on-premises equipment is more complex compared to cloud deployment.

• **Internet and cloud connectivity design:** Evaluate Internet access requirements for uses such as web browsing, cloud application access, or control connections to SD-WAN Control Components in the cloud. Determine how WAN Edge devices will connect to the Internet and the

Technet24

security policies needed to safeguard this traffic. Decide if localized security measures or cloud-based SASE security is required.

• **Security planning:** In addition to securing the Internet access, consider other security aspects, such as encryption, data segmentation for protecting sensitive data, and consistent security policies across the SD-WAN fabric.

• **High availability and disaster recovery:** Design for redundancy at multiple levels (such as link, device, and SD-WAN Control Components) to ensure high availability and reduce downtime. Consider various disaster recovery scenarios to ensure that the network supports business continuity.

• **Compliance and regulatory adherence:** Ensure that the design aligns with industry-specific compliance and regulatory requirements, especially if dealing with sensitive data.

• **SD-WAN overlay tunnels:** Calculate the total number of data tunnels per router based on the deployment scale, color scheme, and overlay topology choices.

• **Cisco SD-WAN platforms:** Select WAN Edge routers that meet your current and future network requirements. Consider factors such as router type (physical or virtual), required throughput, number and type of interfaces, and number of supported data tunnels. Verify that your chosen WAN Edge routers support all the features that you plan to implement; keep in mind that Cisco ThousandEyes and advanced security features have additional memory

requirements. In addition, ensure that WAN Edge devices are properly licensed.

By including these considerations in your design process, you can build a Cisco SD-WAN network that is functional, robust, and scalable. But remember: Every network is unique, and these considerations should be tailored to fit the specific needs and goals of your organization.



Cisco SD-WAN Multi-Region Fabric

Large SD-WAN fabrics may span multiple geographic regions, each with unique routing infrastructure, service providers, and traffic policies. While Cisco SD-WAN is fully capable of supporting such networks, configuring them can pose significant challenges.

To address complexities and streamline deployment, Cisco introduced a new architectural approach known as Multi-Region Fabric (formerly referred to as Hierarchical SD-WAN) that enables organizations to build unified, scalable, and resilient networks spanning multiple locations while ensuring consistent connectivity, security, and management.

Multi-Region Fabric introduces the capability to divide the overlay network into the following components:

• **Core region:** Designated as region 0, this network consists of border

Technet24

routers (BRs in Figure 5-4) that connect to regional overlays and to each other.

• **Access regions:** These networks consist of WAN Edge routers that connect to other WAN Edge routers within the same region. In addition, they can connect to core region border routers assigned to the region.

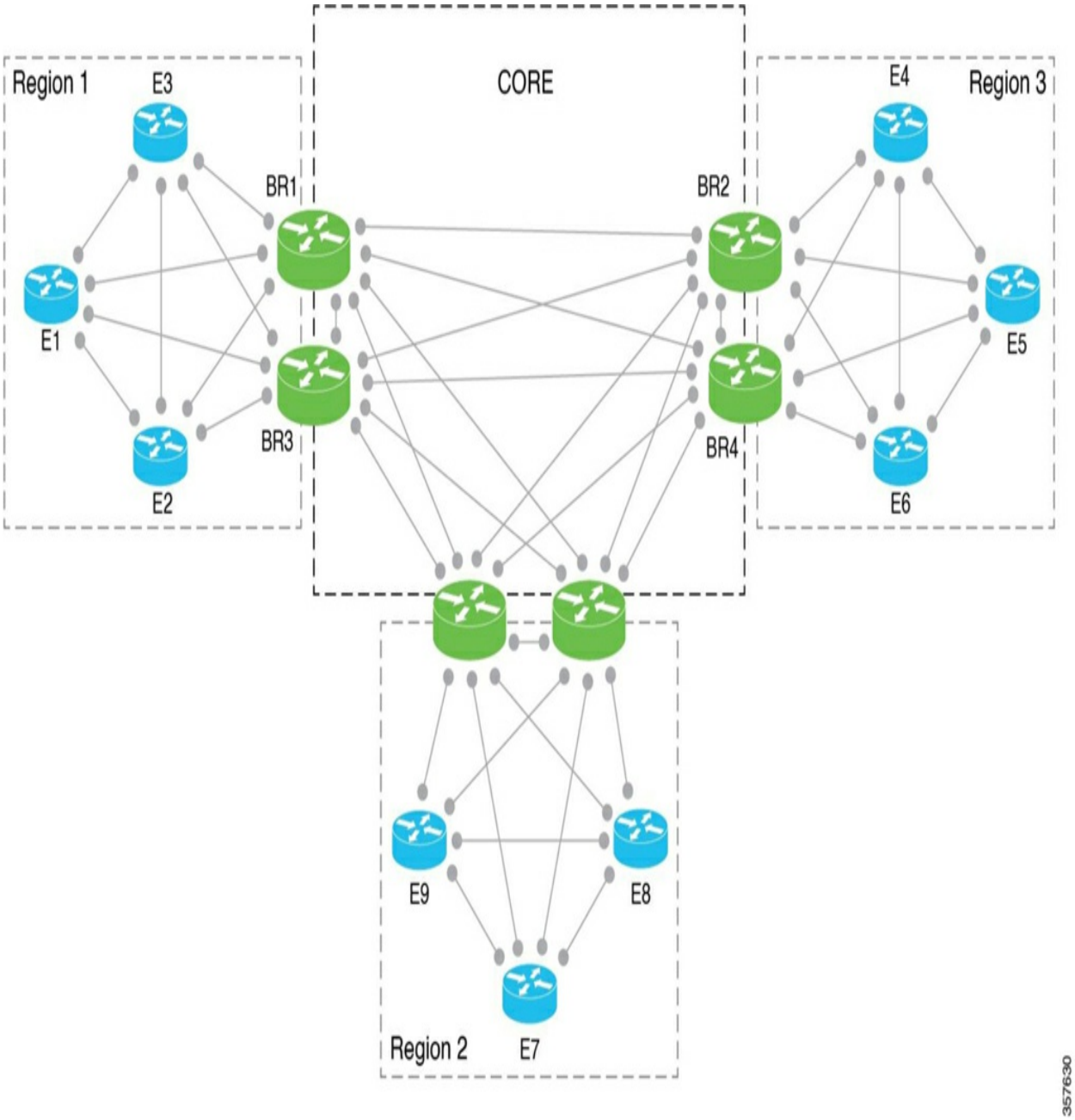


Figure 5-4 *Cisco SD-WAN Multi-Region Fabric*

The division into regions creates a distinction between intra-region traffic and inter-region traffic:

• **Intra-region traffic:** Within a region, WAN Edge routers by default

Technet24

establish direct tunnels to other routers, allowing traffic to flow directly between source and destination devices without intermediate routing.

• **Inter-region traffic:** Conversely, for inter-region communications, WAN Edge routers in one region do not form direct tunnels to routers in different regions. Instead, they rely on core border routers to forward the traffic to the core border routers assigned to the target region. Subsequently, the traffic is directed to WAN Edge routers within the target region. In summary, inter-region traffic traverses at least three tunnels between the source device and the destination device to reach it.

Given the independence of routing within each region in a Multi-Region Fabric, it is possible to customize some aspects of the SD-WAN architecture in each region, including:

• **Topology:** In the core region, it is recommended to use a full-mesh tunnel topology within the overlay network. This setup ensures optimal traffic forwarding between regions by allowing each border router in the core region to establish direct tunnels to every other border router.

Access regions, however, may adopt any suitable topology based on their specific requirements.

• **Transport selection:** The core region’s role is to ensure high performance and reliability for long-distance connectivity between distant geographic regions. Hence, premium transport services like premium WAN circuits,

cloud backbones, or SDCI backbones are typically used in the core region.

In access regions, organizations may use the best available transport options based on their specific needs and constraints.

• **Encapsulation:** In scenarios where the core region infrastructure is secure and does not require encryption, GRE encapsulation may be used for tunnels within the core region to provide better throughput. Conversely, if end-to-end encryption of inter-region traffic is necessary, IPsec encapsulation can be used to ensure secure transmission across the network.

However, the most significant advantage of Multi-Region Fabric is simplified policy design, incorporating functionalities that would typically require complex centralized control policies:

• Traffic between regions is automatically routed via designated border routers. When multiple border routers are present in a region, you can use *router affinity groups* to indicate the order of preference for the routers.

• *Secondary regions* allow direct connection between non-border WAN Edge routers in different regions, bypassing the core region. This is useful for high-volume traffic or applications that cannot tolerate additional latency introduced by routing via the core region.

• Establishing paths between sites using disjointed providers is straightforward: Sites only require reachability to border routers, and then the

Technet24

core region’s network facilitates connectivity between them. However, in some scenarios, it may be advantageous to offload this transit functionality from border routers due to factors such as traffic load, deployment location, or other design considerations. *Transport gateways* provide an alternative option to facilitate connectivity between disjointed providers. WAN Edge routers can be strategically deployed as transport gateways in access regions at points where different transport types converge. There they can handle translation and encapsulation of traffic between disparate transports, ensuring seamless communication across the network.

Note

Router affinity, as a component of symmetric routing, and transport gateway features are available for standard, non-Multi-Region Fabric deployments beginning with Cisco Catalyst SD-WAN Version 20.12/IOS XE Version 17.12.

It is important to note that WAN Edge routers that are not configured for Multi-Region Fabric cannot establish connections with routers that are configured for it. This means that if you decided to use Multi-Region Fabric, all devices in the network must be configured to use it. Due to this restriction, enabling Multi-Region Fabric for an existing SD-WAN network could potentially disrupt connectivity between devices within the network, and so it needs to be planned carefully.



Cisco SD-WAN Control Components Design

As previously discussed, the Cisco Catalyst SD-WAN Control Components are SD-WAN Manager (formerly vManage) for network management, SD-WAN Controller (formerly vSmart) for the control plane, and SD-WAN Validator (formerly vBond) for the orchestration plane.

Early in any Cisco SD-WAN deployment, you need to deploy and configure SD-WAN Control Components and then set up of data center and remote sites. The sequence is crucial because without a functional control plane in place, WAN Edge devices are unable to join the SD-WAN fabric and establish data plane connectivity.

The SD-WAN Control Components, which are software-based, offer deployment flexibility and can be implemented in several ways:

• **Cisco cloud- hosted:** This is the recommended model, and it has been widely adopted due to its ease of deployment, scalability, reduced operational overhead, and cost efficiency. Cisco provisions SD-WAN Control Components in AWS or Azure Cloud, ensuring scale and redundancy, and enabling management of backups and disaster recovery. Customers are granted access to SD-WAN Control Components for network configuration and monitoring. Customers can submit fabric provision requests and manage their deployments through the Cisco Catalyst SD-WAN portal

Technet24

(https://ssp.sdwan.cisco.com), a cloud infrastructure automation tool for Cisco Catalyst SD-WAN. The SD-WAN portal also offers additional functions, such as controller monitoring, fabric maintenance, and more.

• **Managed service provider (MSP) or partner- hosted:** This model is similar to the Cisco cloud-hosted model, but in this case, the MSP or partner is typically responsible for provisioning of the SD-WAN Control Components, backups, and disaster recovery.

• **On-premises deployment in a private cloud or data center owned by an organization:** In this model, customers assume the responsibility of provisioning SD-WAN Control Components, managing backups, and disaster recovery. Some customers, especially those in the finance or government sectors, may opt for on-premises deployments for compliance and security reasons.

• **Deployment in a public cloud owned by your organization:** This model is similar to the on-premises option, but customers leverage their own public cloud infrastructure, eliminating the need for investing in their data center infrastructure.

Figure 5-5 highlights customer responsibilities for each of these SD-WAN Control Component deployment options.

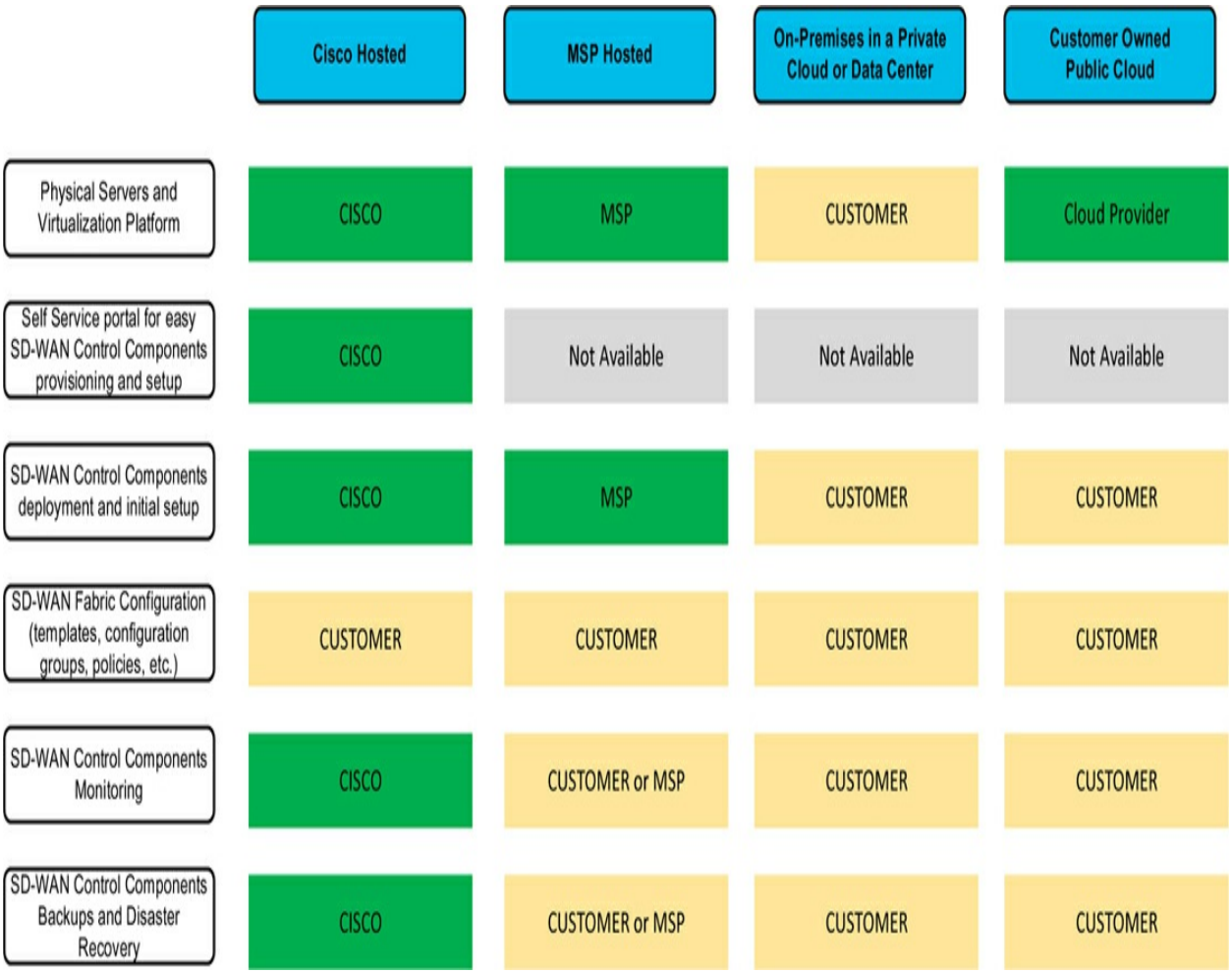


Figure 5-5 *Customer Responsibilities for Control Component Deployment Options*

All the deployment options involve the following common design considerations:

• **Connectivity channels:** Ensure that all SD-WAN Control Components and WAN Edge routers are reachable to each other via their VPN 0 interface.

• **IP address assignment:** Choose what kind of IP addresses (private or publicly routable) to assign to SD-WAN Control Components. When using

Technet24

private IP addresses with NAT, SD-WAN Validator learns about both private and public addresses of the SD-WAN Controller and SD-WAN Manager Control Components and sends this information to WAN Edge routers.

• **Transport colors:** Choose transport colors assigned to SD-WAN Control Components. The type of color makes a difference since with public colors, WAN Edge routers will always connect to the public (post-NAT) IP addresses of Control Components, while with private colors on both sides, WAN Edge routers will try to connect to private (pre-NAT) IP addresses.

• **Placement:** Decide where to strategically place SD-WAN Control Components in the network topology. Ensure that any firewall in the path allows communication between SD-WAN Control Components, as well as between WAN Edge routers and SD-WAN Control Components.

Note that with the Cisco cloud-hosted model, these considerations are addressed during provisioning. Connectivity channels are ensured during provisioning; SD-WAN Control Components are assigned private IP addresses and are mapped one-to-one via NAT to public IP addresses; and a public default color is assigned to transports (although it can be changed after provisioning). In addition, with SD-WAN Control Components hosted in the cloud, their persistent public IP addresses are assigned during provisioning and made known via the Cisco Catalyst SD-WAN portal so firewall rules can be properly configured.

Cloud-Hosted SD-WAN Control Components Deployments

Cloud-hosted deployments offer the advantages of ease of deployment and scalability, requiring only Internet accessibility. However, this convenience can turn into a disadvantage when SD-WAN fabric lacks Internet transports. Even if one Internet transport is available at a remote site, it introduces a vulnerability in terms of control connectivity redundancy.

Let’s explore typical design scenarios for control connectivity, using a simple SD-WAN network with Internet and MPLS transports as an example.

With Option A, illustrated in Figure 5-6, MPLS transport provides native Internet reachability through some extranet connection. This configuration enables WAN Edge routers to establish direct connections to SD-WAN Control Components and form data tunnels using both transports. For WAN Edge routers to be able to reach the SD-WAN Control Components, they require either a default route or specific routes toward the publicly routable IP addresses of the SD-WAN Control Components. These routes can be configured statically or received dynamically via a routing protocol from the MPLS customer edge (CE) router. The choice between these approaches depends on the specific network requirements and the desired level of control.

Technet24

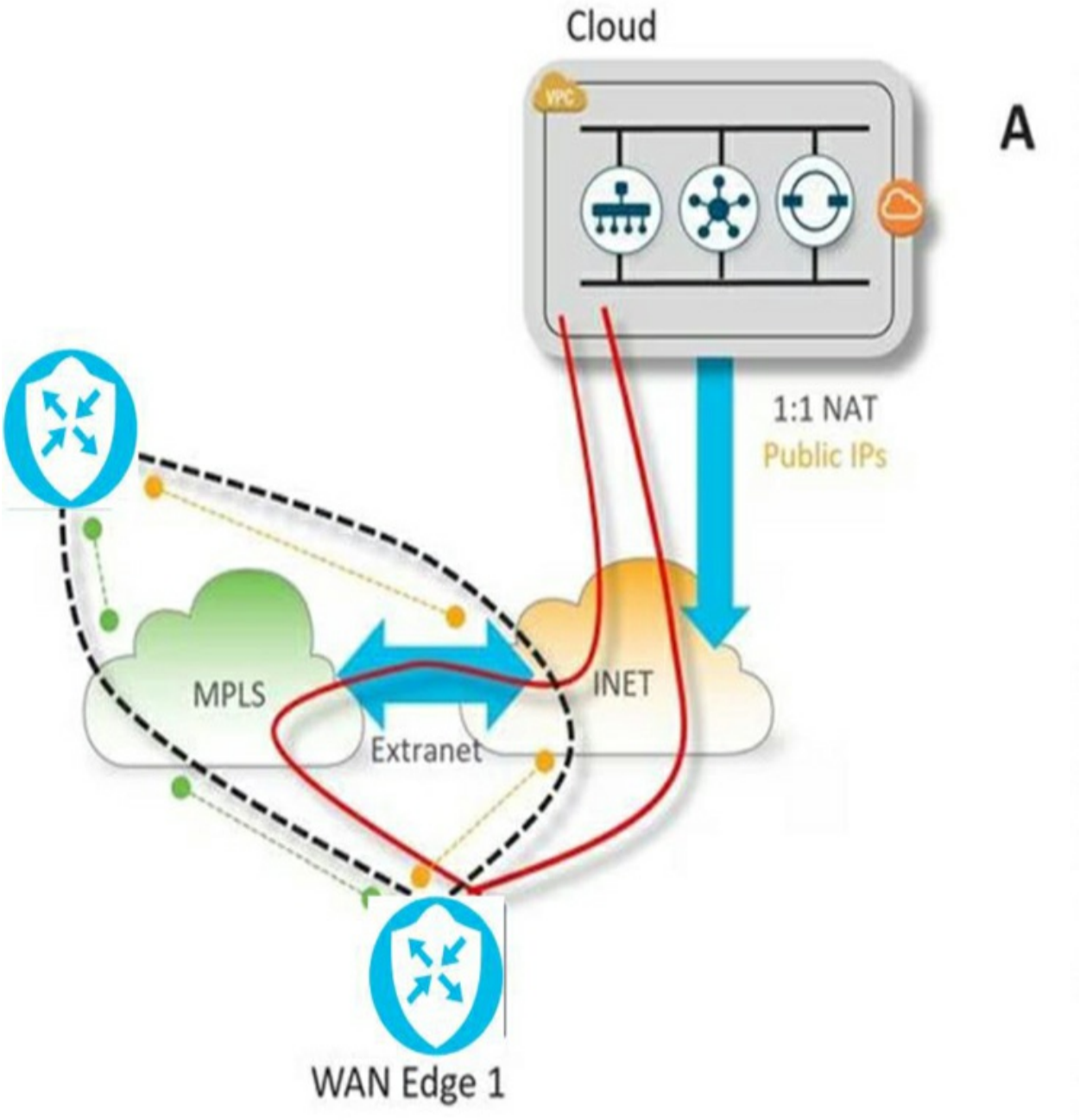


Figure 5-6 *Cloud-Hosted SD-WAN Control Components, Option A*

In Option B, illustrated in Figure 5-7, the MPLS transport lacks direct Internet connectivity. Instead, control traffic is routed via a data center (or a regional hub) that is connected to both transports. In this configuration, data center routers may need to advertise either a default route or publicly routable

IP addresses of SD-WAN Control Components into the MPLS network, depending on the specific network requirements.

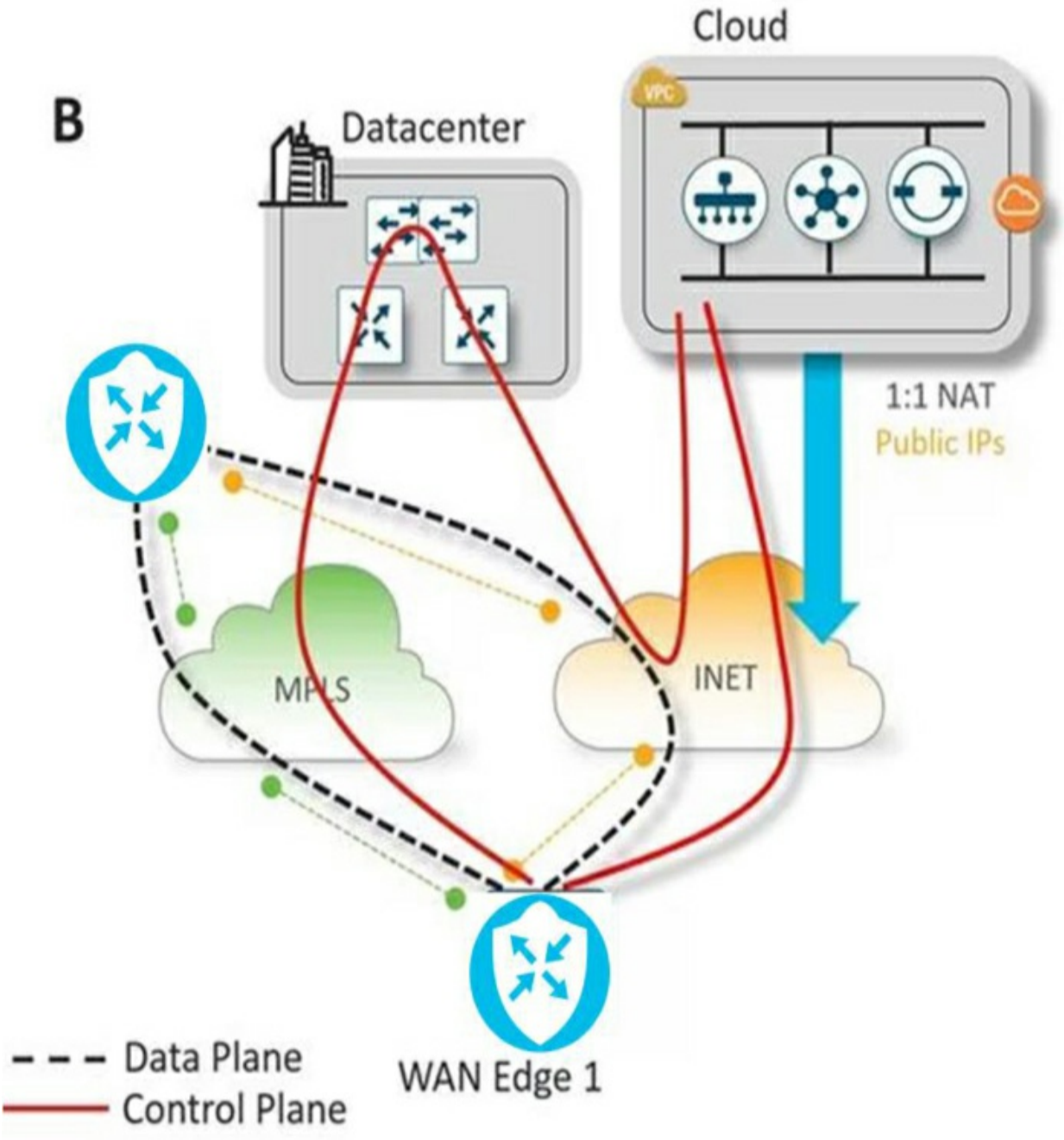


Figure 5-7 *Cloud-Hosted SD-WAN Control Components, Option B*

In Option C, shown on Figure 5-8, the MPLS transport does not provide

Technet24

Internet access. Consequently, WAN Edge routers can only establish connections to SD-WAN Control Components over the Internet transport. Once this connection is built, complete control information, including MPLS TLOCs, is received, enabling the WAN Edge to construct data tunnels over the MPLS transport. However, this setup lacks control plane redundancy. In the event of Internet transport failure, SD-WAN functionality at the site will degrade (for example, the WAN Edge will not receive routing updates), and the site will eventually go down when the IPsec rekey timer expires, the Overlay Management Protocol (OMP) graceful restart timer lapses, or the WAN Edge router is rebooted. This option is therefore the least preferred.

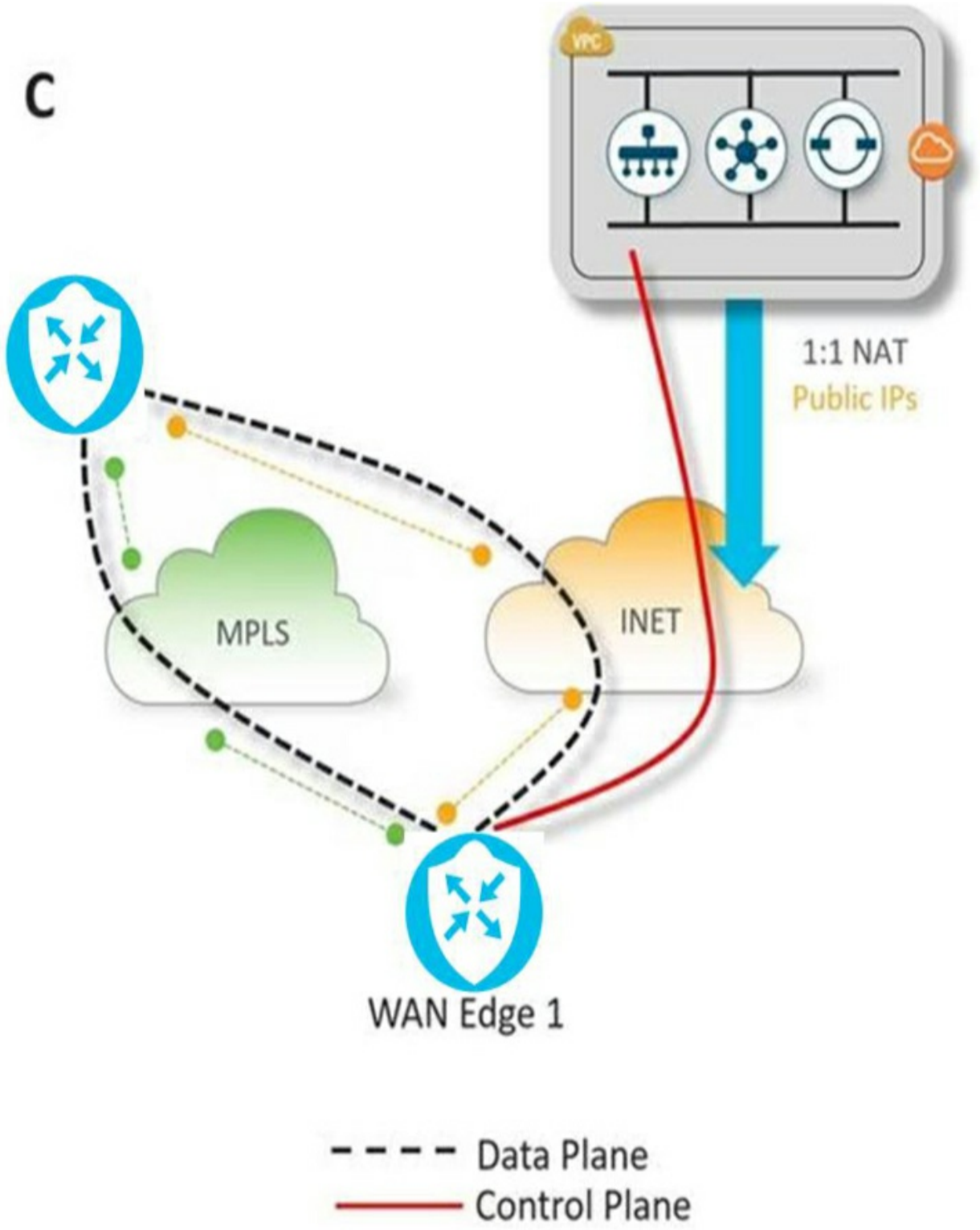


Figure 5-8 *Cloud-Hosted Control Components, Option C*

It is important to note that with default settings, WAN Edge routers will

Technet24

attempt to establish control connections to SD-WAN Control Components over the MPLS transport. If such an attempt is unsuccessful, the router will not proceed to form data tunnels over it, and SD-WAN will not use the transport. To address this, you should set the Maximum Control Connections option for the MPLS tunnel interface to 0. This tells the WAN Edge router that this TLOC is not expected to form control connections but still can build data tunnels.

In a cloud-hosted environment, SD-WAN Control Components are usually placed into a subnet behind a virtual gateway. Each SD-WAN Control Component is assigned a private IP address, and the cloud virtual gateway uses one-to-one NAT to translate that address into a unique public IP address that is accessible from the Internet.

A public color is assigned to the tunnel interfaces of SD-WAN Manager and SD-WAN Controller. This ensures that WAN Edge devices use public IP addresses of SD-WAN Control Components when communicating with them. The actual value of the color is not important and is often left unassigned, resulting in the use of the default color, which is public. SD-WAN Validator is not configured with a tunnel interface, so the concept of color does not apply to it.

When SD-WAN Manager and SD-WAN Controller communicate with SD-WAN Validator, it is important for this interaction to use public IP addresses, even if all Control Components are in the same subnet. This is necessary

because SD-WAN Validator needs to learn public IP addresses of other Control Components and relay this information to WAN Edge routers joining the SD-WAN overlay.

To meet this requirement, SD-WAN Manager and SD-WAN Controller are configured with the public IP address of SD-WAN Validator. This way, when they attempt to connect to SD-WAN Validator, the traffic is directed to a virtual gateway. The gateway then translates the private source IP addresses to the public IP addresses of SD-WAN Control Components, ensuring that SD-WAN Validator perceives the traffic as originating from the NAT public IP addresses of SD-WAN Manager and SD-WAN Controller. When SD-WAN Validator responds, it sends the returning traffic via the same virtual gateway where the reverse IP address translation process happens, achieving traffic symmetry.

SD-WAN Manager and SD-WAN Controller can communicate with each other both using private and public IP addresses. This is dictated by their site ID assignment: If the site IDs are the same, SD-WAN Manager and SD-WAN Controller communicate using private IP addresses. On the other hand, if the site IDs are different, Control Components communicate with each other via their NAT public IP addresses. This approach is used in Cisco cloud-hosted deployments.

On-Premises SD-WAN Control Components Deployments

Setting up SD-WAN Control Components on premises involves several

Technet24

considerations related to NAT, color usage, and the choice between public and private IP addresses. This section explores design options using a sample deployment featuring both public (Internet) and private (MPLS) circuits. In this scenario, the Internet TLOC is assigned the public color **biz-internet**, while the MPLS TLOC is assigned the private color **mpls**. In this configuration, control connections are established over both transports, as illustrated in Figure 5-9.

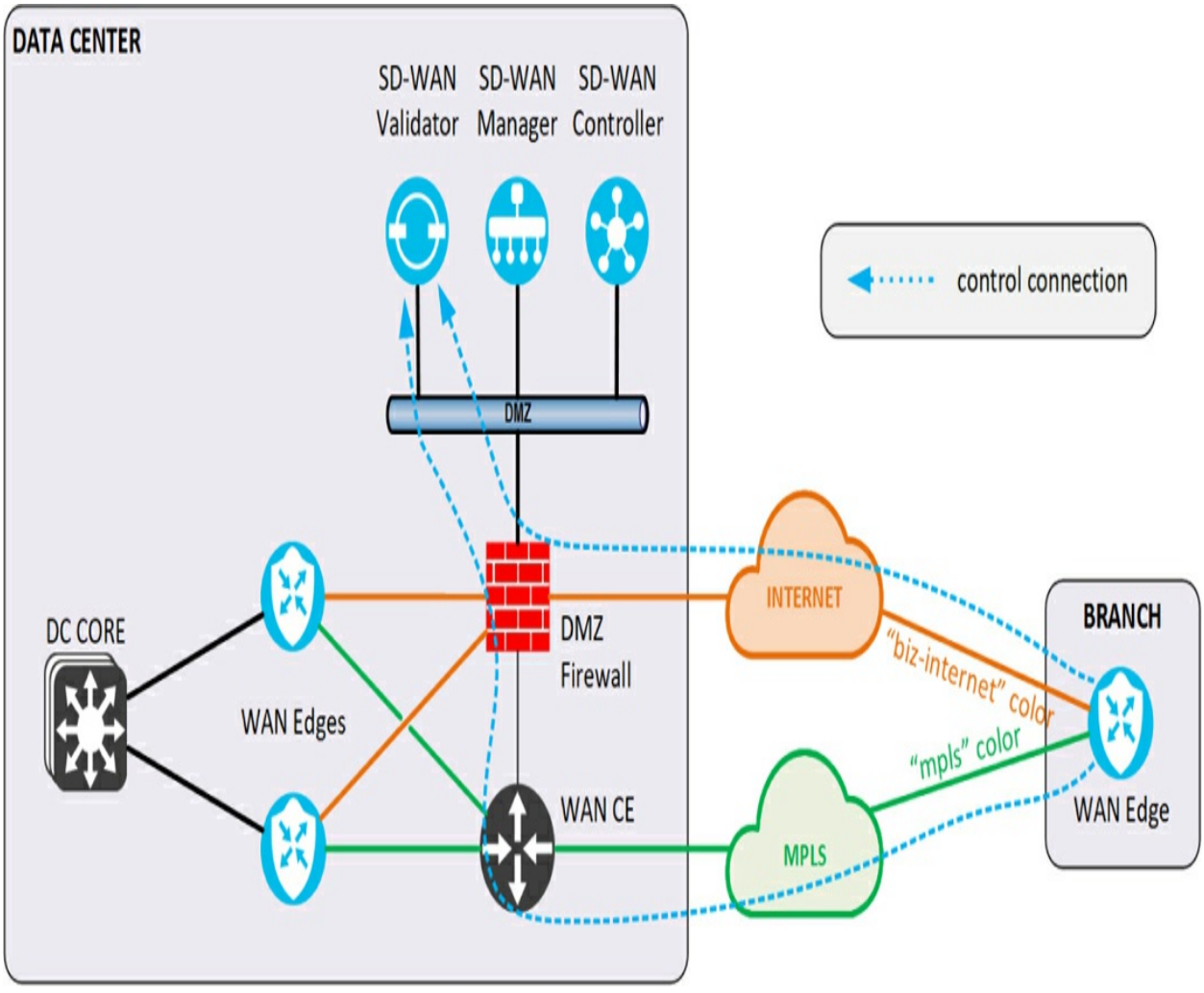


Figure 5-9 *On-Premises SD-WAN Control Components Deployment Example*

In this setup, all SD-WAN Control Components reside within a dedicated DMZ inside the data center. However, alternative deployment scenarios are possible, as long as they ensure connectivity between SD-WAN Control Components in VPN0 and allow SD-WAN Validator to discover the public (post-NAT) IP addresses of SD-WAN Manager and SD-WAN Controller.

To provide consistent connectivity to SD-WAN Control Components over both public and private transports, several IP addressing options are available:

• Public IP addresses:

• SD-WAN Control Components are assigned public routable IP addresses.

• Since public IP addresses are unknown within the MPLS cloud, the IP addresses of Control Components (or a default route) must be advertised into the MPLS network from the data center. This ensures that the Control Components are reachable by WAN Edge routers.

• Private IP addresses, mapped via one-to-one NAT to public IP addresses:

• As in the previous case, post-NAT public IP addresses of SD-WAN Control Components need to be advertised into MPLS.

• SD-WAN Controller and SD-WAN Manager are configured with a public color, so WAN Edge routers communicate using their public IP addresses

Technet24

over both transports.

• Much as in cloud-hosted deployments, to facilitate discovery of their public IP addresses, SD-WAN Manager and SD-WAN Controller communicate with SD-WAN Validator using its public IP address. Meanwhile, connectivity between SD-WAN Manager and SD-WAN Controller uses private IP addressing because both are configured with the same site ID, as shown in Figure 5-10.

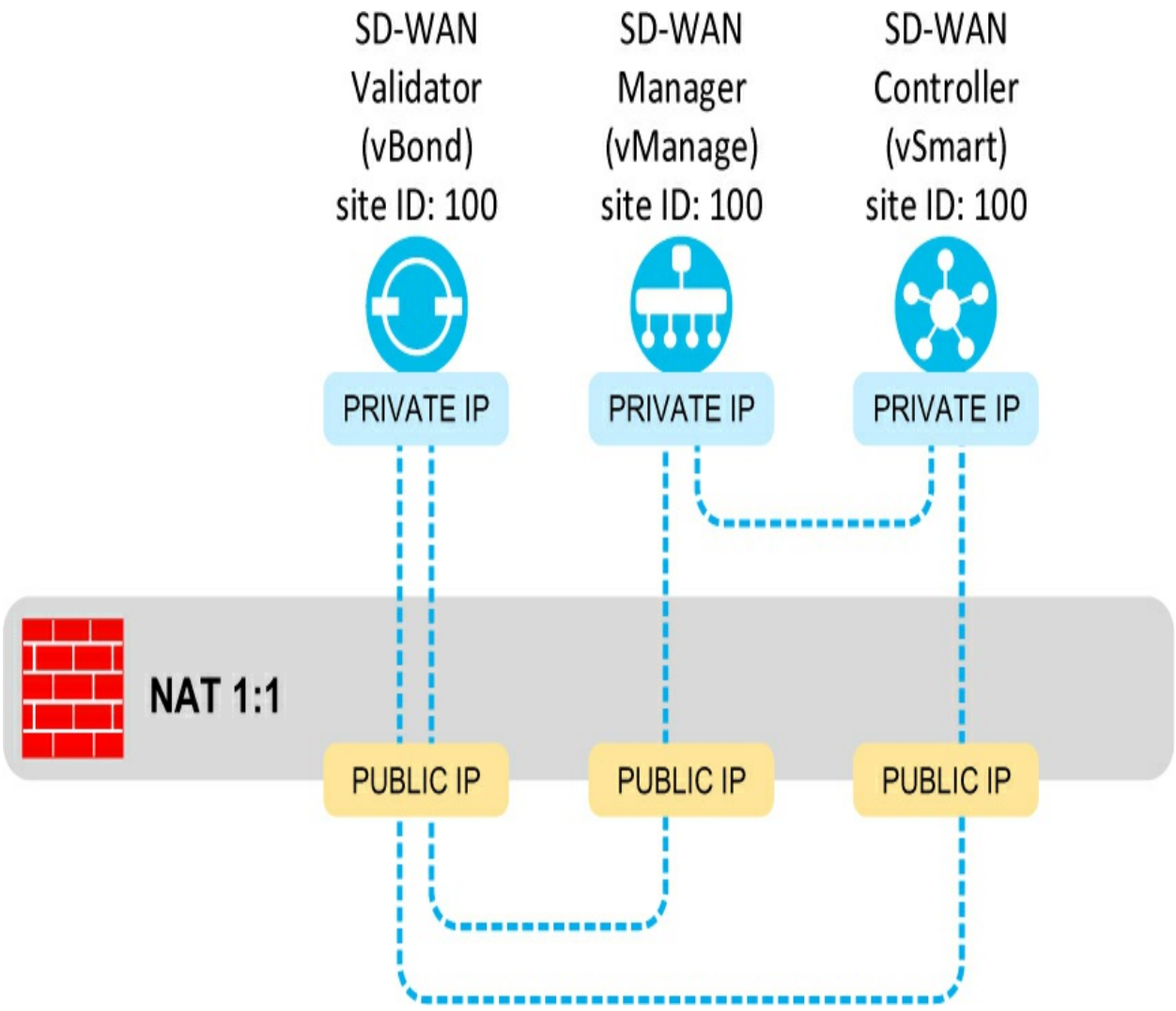


Figure 5-10 *On-Premises SD-WAN Control Components with Private IP*

Note

The firewall should properly implement hairpin NAT so that traffic from SD-WAN Control Components is correctly translated when traversing the firewall, even if the destination is in the same subnet. If this functionality is not supported, SD-WAN Validator may need to be implemented in its own dedicated DMZ.

• Mixed public and private IP addresses:

• SD-WAN Control Components are assigned private IP addresses that are mapped via NAT to public IP addresses on public transport only.

• The SD-WAN Validator DNS name is set to resolve to both its public and private IP addresses.

• SD-WAN Controller and SD-WAN Manager are configured with a private color.

• Over the Internet transport, WAN Edge devices communicate with the public IP addresses of SD-WAN Control Components because routers are configured with the public **biz-internet** color. (Remember that when one side of a control connection uses a private color and another uses a public color,

Technet24

the control connection is formed using the public IP addresses of devices).

• Over the MPLS transport, WAN Edge routers use the private IP addresses of SD-WAN Control Components since both routers and Control Components use private colors.

As you can see, these deployment options offer enough flexibility to tailor the implementation based on the requirements and constraints of specific networks.

SD-WAN Control Components Redundancy and High Availability

In any network setup, a standard best practice is to install redundant components at all levels, including duplicate routers, power supplies, and other components, as well as backup network connections. Similarly, it is important to provide high availability for SD-WAN Control Components. Networks designed for resilience should include redundant SD-WAN Manager, SD-WAN Controller, and SD-WAN Validator instances. By deploying duplicate instances of these SD-WAN Control Components, an organization can ensure uninterrupted operations even in the face of hardware failures, outages, or other unforeseen issues.

SD-WAN Manager Design

To enhance the availability of SD-WAN Manager, you can deploy it in one of the following ways:

• **SD-WAN Manager cluster:** A cluster consists of three or six SD-WAN Manager servers, all operating in active mode. This setup provides protection against a single node failure.

• **Standalone SD-WAN Manager in active/standby mode:** You can use one SD-WAN Manager instance in active mode as the primary with another instance in standby mode as a backup.

• **SD-WAN Manager cluster in active/standby mode:** You can set up a primary SD-WAN Manager cluster in active mode, with another SD-WAN Manager cluster in standby mode to act as a backup.

When deployed in a cluster, SD-WAN Manager servers assume specific roles based on the services they run. This distribution of roles enables the cluster to balance the workload efficiently and share information among SD-WAN Manager servers. The role of a server in a cluster is determined by its persona. An SD-WAN Manager server can assume any of the following personas:

• **Compute+Data:** This persona runs all services necessary for SD-WAN Manager, including application server, statistics database, configuration database, messaging server, and coordination server.

• **Compute:** This persona runs all services except statistics database.

• **Data:** This persona runs application server and statistics database services.

Technet24

The persona for SD-WAN Manager is selected the first time that the server boots up after the SD-WAN Manager installation and cannot be changed. When deciding on the persona to configure for a server, note that the following deployments of nodes are the only officially supported combinations for an SD-WAN Manager cluster:

• Three Compute+Data nodes

• Three Compute+Data nodes and three Data nodes

If your network requires a different combination of nodes, it is advisable to contact your Cisco representative to validate the deployment.

When planning an SD-WAN Manager cluster, keep the following design considerations in mind:

• A third interface is required in VPN 0 for communication and synchronization among SD-WAN Manager servers in the cluster. It is recommended to use a 10 Gbps interface for this purpose.

• Database replication between cluster members requires 4 ms or less delay between cluster members. To achieve this, it’s advisable to have all cluster members located at the same site.

• All control sessions between the WAN Edge routers and SD-WAN Manager application servers are load balanced to optimize performance. A

single SD-WAN Manager instance can potentially manage up to 1000 to 1500 routers, depending on the load. If your network exceeds this device count, or if the load is high for a single server, you need to deploy an SD-WAN Manager cluster.

• A WAN Edge router establishes a single connection to SD-WAN Manager over one of the transports at any given time. You can specify the preferred transport by using the vManage Connection Preference setting under the tunnel interface on a WAN Edge, where a higher value indicates a higher preference. For instance, you might use a preference of 5 (default) for a high-bandwidth Internet link, a value of 3 for a lower-bandwidth MPLS link, and a value of 1 on a metered cellular interface to minimize the costs. A 0 value indicates that the tunnel interface will never be used to connect to the SD-WAN Manager. At least one tunnel interface must have a nonzero value.

It’s important to recognize that the main purpose of an SD-WAN Manager cluster is scalability. While a cluster offers a level of redundancy against a single SD-WAN Manager failure, it does not protect against cluster-level failures. To achieve full redundancy, you need to deploy either a backup SD-WAN Manager or an SD-WAN Manager cluster in standby mode.

In common disaster recovery scenarios, an active SD-WAN Manager or SD-WAN Manager cluster is located in one data center site along with at least one SD-WAN Validator and SD-WAN Controller, and in a second data center, a standby (inactive) SD-WAN Manager or SD-WAN Manager cluster

Technet24

is deployed, also with at least one active SD-WAN Validator and SD-WAN Controller. The following disaster recovery methods are available:

• **Manual:** The backup SD-WAN Manager server or cluster is kept shut down in a cold standby state. Regular backups of the active database are taken manually, and if the primary SD-WAN Manager or SD-WAN Manager cluster goes down, the standby is brought up manually, and the backup database restored on it. This approach is applicable to both standalone SD-WAN Manager and SD-WAN Manager cluster deployments.

• **Administrator-triggered failover:** This is the recommended approach. In this method, the standby server or cluster is in a warm standby mode, and data is replicated automatically between the primary and secondary SD-WAN Manager instances at regular intervals. When needed, a switchover to the secondary SD-WAN Manager instance is performed manually from the GUI with a simple button click.

SD-WAN Controller Design

In a Cisco SD-WAN network designed for high availability, SD-WAN Controller redundancy is achieved by adding extra SD-WAN Controllers that operate in an active/active fashion. A network can have multiple SD-WAN Controllers (up to 12 have been tested), and each WAN Edge router, by default, connects to 2 of them at a time.

While the configurations on all SD-WAN Controllers must be functionally

similar, the control policies must be completely identical. This ensures that, at any time, all WAN Edge devices receive consistent views of the network. The simplest and recommended approach to achieve this is by using SD-WAN Manager to manage SD-WAN Controllers and configure control policies rather than by configuring SD-WAN Controllers through the CLI.

SD-WAN Controllers assume the central control role within the network and maintain it by establishing the following connections:

• A full mesh of permanent DTLS/TLS control connections and OMP sessions among all SD-WAN Controllers to advertise routes, TLOCs, services, policies, and encryption keys. This exchange of information allows the SD-WAN Controllers to remain fully synchronized.

• A permanent DTLS connection to each SD-WAN Validator from each SD-WAN Controller's CPU core. These connections enable SD-WAN Validators to track the presence and operational status of SD-WAN Controllers.

• Permanent DTLS/TLS control connections and OMP sessions with the WAN Edge routers.

These connections keep SD-WAN Controllers constantly aware of any changes in the network topology. In the event of an SD-WAN Controller failure, the remaining SD-WAN Controllers seamlessly take over control of the network. They handle WAN Edge devices joining the network and continue sending route updates to the routers.

Technet24

Let’s review in more detail how WAN Edge routers interoperate with SD-WAN Controllers. When a WAN Edge device comes into the network, it receives a list of all available SD-WAN Controllers from SD-WAN Validator during the registration process; it selects which ones to use and designates them “assigned”.

In cases where there are more SD-WAN Controllers in the network than a WAN Edge device is allowed to connect to (two by default), the router selects a subset of SD-WAN Controllers as “assigned.” The selection is based on a hash that includes the system IP address of the SD-WAN router. This makes this process deterministic, meaning that WAN Edge routers consistently choose the same set of “assigned” SD-WAN Controllers while introducing enough randomness to ensure that different WAN Edge devices select different “assigned” SD-WAN controllers. This ensures that control connections and OMP sessions are load-balanced across all SD-WAN Controllers.

Note

To maximize the efficiency of the load-balancing among SD-WAN Controllers, it is recommended to use sequential numbers when assigning system IP addresses to the WAN Edge routers.

Control connections from WAN Edge routers to an SD-WAN Controller are

also load-balanced across different cores on the same SD-WAN Controller. When the SD-WAN Validator sends the list of available SD-WAN Controllers to a WAN Edge router as part of the registration response, it includes only one of the available SD-WAN Controller cores. The SD-WAN Validator determines which core to send within a given registration response to balance control connections across all the cores of a given SD-WAN Controller.

Subsequently, WAN Edge routers establish permanent DTLS/TLS control connections and OMP sessions with the “assigned” SD-WAN Controllers. At this point, WAN Edge devices consider themselves to be in equilibrium if all the expected control connections and OMP sessions are successfully established.

Figure 5-11 illustrates this process. In this example, the WAN Edge router selected SD-WAN Controllers 1 and 3 as “assigned.” However, it could have chosen another pair, such as SD-WAN Controllers 1 and 2 or SD-WAN Controllers 3 and 4, if it had a different system IP address.

Technet24

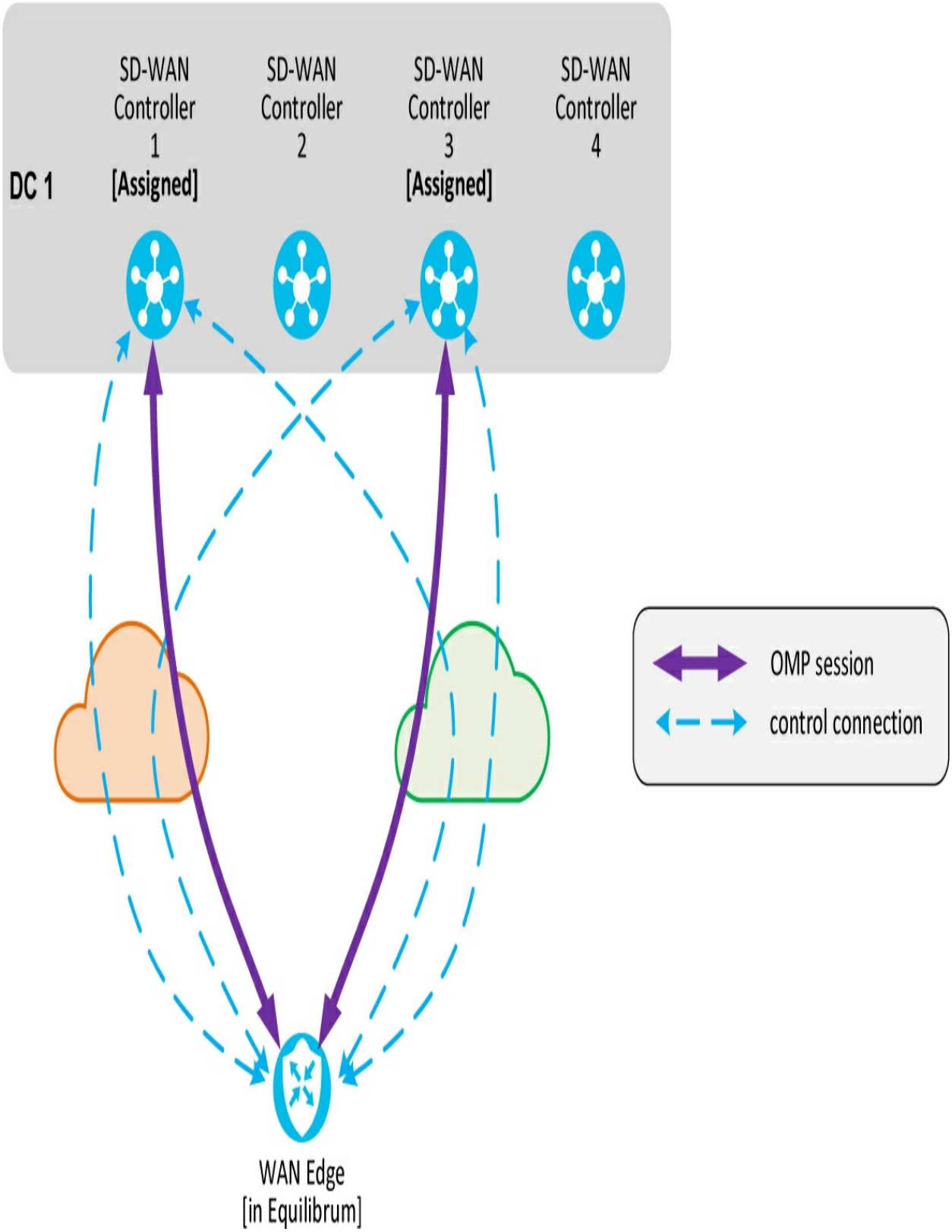


Figure 5-11 *WAN Edge Control Connections to SD-WAN Controllers*

By default, each WAN Edge router establishes DTLS/TLS control connections to two SD-WAN Controllers over each TLOC. This is controlled at the WAN transport tunnel interface level by the Maximum Control Connections setting. Similarly, each router establishes OMP sessions to two SD-WAN Controllers by default. This is controlled by the Maximum OMP Sessions setting in the System Configuration section. It is generally recommended to leave these settings unchanged as defaults, to minimize the number of connections made to the SD-WAN Controllers while still maintaining a good level of redundancy.

If an “assigned” SD-WAN Controller becomes unreachable, the WAN Edge device attempts to establish a connection to one of the remaining SD-WAN Controllers. At this time, it will be in the “out of equilibrium” state, continuously trying to reconnect to the unavailable “assigned” SD-WAN Controller to regain equilibrium. This way, when the “assigned” SD-WAN Controller comes back online, it restores all its previous sessions with WAN Edge routers, effectively rebalancing connections among SD-WAN Controllers.

As your network expands and more SD-WAN Controllers are added, you might need better control over which SD-WAN Controllers your WAN Edge routers connect to. Consider the following scenarios:

• In geographically distributed networks, you may prefer to use local SD-WAN Controllers for better performance, with a backup available in more

Technet24

distant regions for redundancy.

• When deploying multiple SD-WAN Controllers in separate data centers, you may want all your WAN Edge devices to connect to one SD-WAN Controller in one data center and one SD-WAN Controller in another during normal operations. In the event of a data center outage, all routers maintain one active connection to an SD-WAN Controller in a surviving data center, avoiding going into graceful restart mode.

• You might want to group devices based on certain criteria (such as site type or size) and allocate specific SD-WAN Controller instances to each group.

SD-WAN Controller affinity allows you to manage scale and exercise control over which SD-WAN Controllers your WAN Edge routers connect to.

To participate in affinity, each SD-WAN Controller must be assigned a controller group identifier with the **controller-group-id** *number* command in the System Configuration section. SD-WAN Controllers in the same data center can have the same controller group identifier or different identifiers:

• If SD-WAN Controllers share the same identifier, WAN Edge routers establish a control connection with any one of them.

• If SD-WAN Controllers have different controller group identifiers, WAN Edge routers can designate one SD-WAN Controller as the preferred and the other(s) as a backup. This design, while offering redundancy among SD-

WAN Controllers in a data center, also provides more granular control and additional fault isolation.

Configuring SD-WAN Controller affinity on a WAN Edge device is a two-part process:

• At the system level, configure a list of affinity groups that the WAN Edge router belongs to in the Controller Groups setting. The router attempts to form connections to controller groups in the list sequentially, starting with the first group, so IDs should be entered in order of preference. It is recommended that this list contain all the controller groups in the overlay network. The total number of established sessions will be limited by the numbers set for the Maximum Control Connections and Maximum OMP Sessions settings.

• Optionally, for each tunnel interface, exclude non-preferred controller groups with the Exclude Controller Group List setting. A WAN Edge router will make a connection to the excluded controller group(s) only as a last resort, when no other preferred controller groups are available. This helps avoid complete loss of connectivity to the SD-WAN network.

Figure 5-12 provides an example of how SD-WAN Controller affinity can be implemented in an SD-WAN deployment featuring two active data centers, with the third one designated for disaster recovery purposes.

Technet24

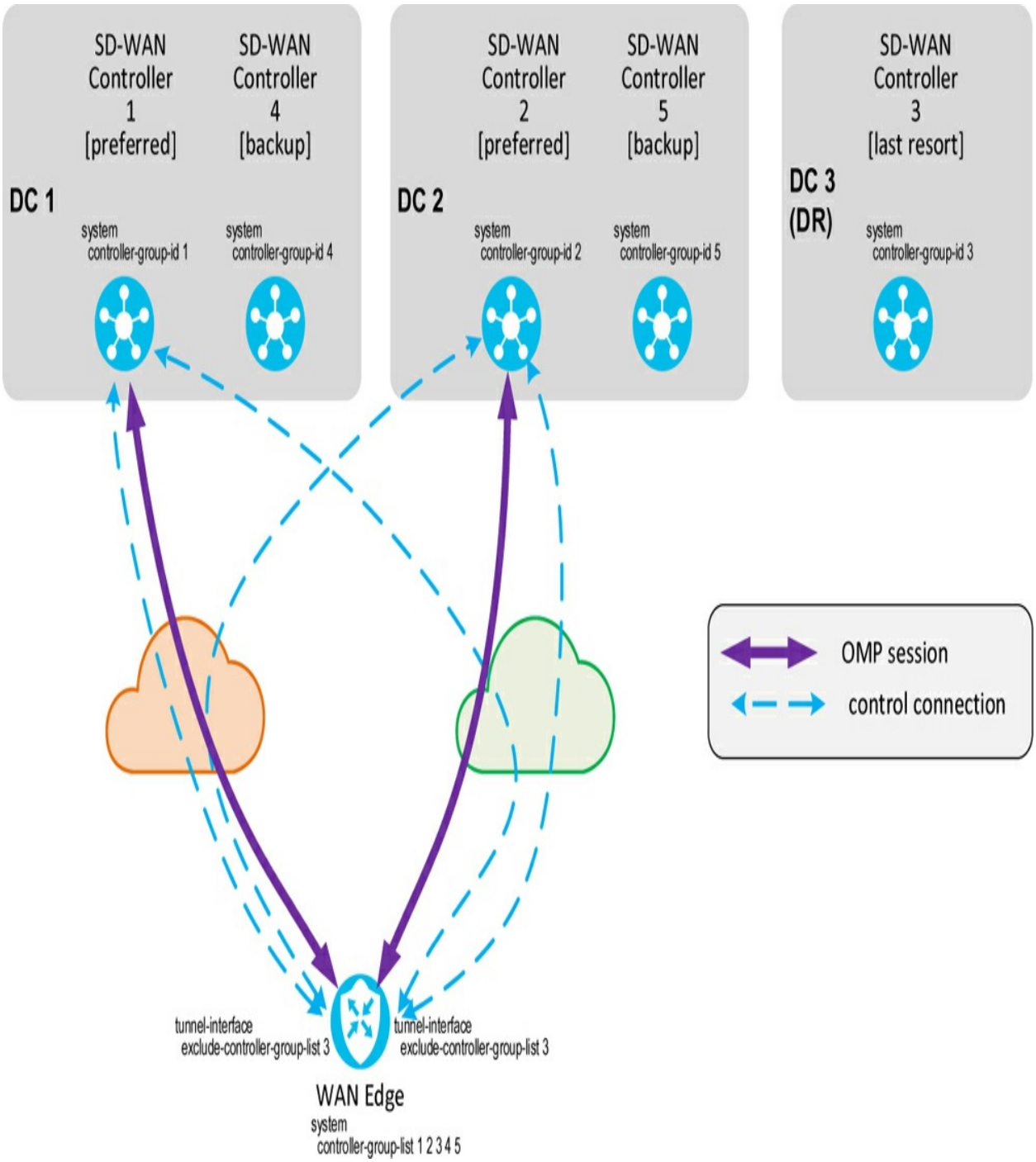


Figure 5-12 *SD-WAN Controller Affinity in a Dual Data Center Deployment with DR*

In this example, a WAN Edge router first attempts to establish connections with SD-WAN Controllers in groups 1 and 2 over each transport. If group 1

becomes unavailable, the router will then attempt connections with the next group in the Controller Group list, which is group 4, bypassing group 3 due to its exclusion with the Exclude Controller Group List setting, and so forth. However, if no other controller groups are available, as a last resort, the router will try to connect to the excluded SD-WAN Controller in group 3 to avoid disconnection from the SD-WAN overlay.

To monitor SD-WAN Controller association and affinity status on a WAN Edge router, you can use the **show sdwan control affinity config** and **show sdwan control affinity status** commands.

Note

For an in-depth examination of SD-WAN Controller affinity design, we recommend referring to the “Cisco Catalyst SD-WAN Large Global WAN Design Case Study” and the “Cisco Catalyst SD-WAN High Availability Configuration Guide” documents, published on the Cisco documentation site. These resources provide comprehensive information beyond what we can cover in this book.

SD-WAN Validator Design

SD-WAN Validator plays a pivotal role in the Cisco SD-WAN overlay network, performing the following essential functions:

Technet24

• Authenticating and validating all SD-WAN Controllers and WAN Edge routers attempting to join the SD-WAN network

• Facilitating the establishment of control connections between WAN Edge routers and the SD-WAN Manager for centralized management and configuration

• Orchestrating control plane connections between the SD-WAN Controllers and WAN Edge devices

• Serving as a STUN server to assist WAN Edge routers in determining their public IP addresses and establishing tunnels between them, even when they are located behind NAT devices

SD-WAN Validator runs as a virtual machine using a vEdge Cloud image, and due to its important role, it is advisable to deploy at least two SD-WAN Validators in your network. This ensures that one of them is always available whenever a router or an SD-WAN Controller is attempting to join the network.

vEdge Cloud learns its role as an SD-WAN Validator through its configuration when the **system vbond** command includes the **local** option. In this command, you also specify the local IP address of the SD-WAN Validator's interface in VPN0. The same command is used on SD-WAN Managers, SD-WAN Controllers, and WAN Edge routers to specify SD-WAN Validator orchestrator(s) for the SD-WAN network. Note that on

WAN Edge routers managed by SD-WAN Manager, this command is not directly configurable. Instead, it is added to the configuration automatically by SD-WAN Manager with the value from the Validator (formerly vBond) setting in the Administration section of SD-WAN Manager.

In a network that has a single SD-WAN Validator, you can specify it either as an IP address or as a DNS name (such as validator.ciscopress.com). However, in networks with two or more SD-WAN Validators, the only option to use them is through a DNS name. The name is resolved to multiple IP addresses, and a router (or SD-WAN Controller) sequentially tries each IP address until a successful connection is established. Note that an external DNS server is not a requirement for an SD-WAN Validator DNS name resolution, as host names can be defined locally, as demonstrated in Example 5-1.

Example 5-1 *SD-WAN Validator Configuration Example (Configuration Snippets)*

```
vBond-1:
system
 vbond 209.165.201.12 local
```

```
vBond-2:
```

Technet24

```
system
 vbond 209.165.201.13 local
```

vSmart-1:

```
system
 vbond validator.ciscopress.com
vpn 0
 host validator.ciscopress.com ip 209.165.201.1
```

BR1-Edge1:

```
system
 vbond validator.ciscopress.com
!
ip host validator.ciscopress.com 209.165.201.11
```

Note

Even when your SD-WAN network has only a single SD-WAN Validator, using a DNS name instead of an IP address is recommended as a best practice. This approach makes configuration more scalable. For example:

• No configuration changes are needed on the network devices if additional SD-WAN Validators are introduced to the network.

• In large geographically distributed networks, regional DNS names can be used to point to local SD-WAN Validator orchestrators.

In a network that has multiple SD-WAN Validators, each operates independently and is not even aware of the presence of other SD-WAN Validators. If one SD-WAN Validator fails, the remaining orchestrators simply continue to operate, handling all requests from Cisco devices attempting to join the network. As long as at least one SD-WAN Validator is operational in the domain, the SD-WAN network can continue operating without interruption because WAN Edge routers and SD-WAN Controllers can still locate each other and join the network.

From the control plane perspective, each SD-WAN Validator maintains permanent connections to each SD-WAN Manager and SD-WAN Controller. This ensures an up-to-date view of the network, preventing SD-WAN Validator from providing the IP address of an unavailable control component to devices joining the network. There are no control connections between SD-WAN Validators themselves, and no state is kept between them.

Note

Technet24

It is commonly thought that WAN Edge routers only establish transient control connections to SD-WAN Validator orchestrators when they join the network. However, there is a scenario when these connections become persistent.

When WAN Edge routers are unable to connect to at least one of their “assigned” SD-WAN Controllers due to issues like downtime, they enter the “out of equilibrium” state. In response, routers establish a persistent DTLS control connection from each of their TLOC to an SD-WAN Validator instance. This helps the WAN Edge devices determine whether the “assigned” SD-WAN Controller instances have been administratively removed from the network or are merely temporarily unavailable, meaning that routers should continue trying to connect to them. This strategy ensures that if the SD-WAN Controllers are down only temporarily, WAN Edge routers will reconnect to the same SD-WAN Controller instances once they are eventually back online. This approach prevents the permanent shift of load to a smaller number of SD-WAN Controller instances over time, which could happen in large-scale deployments due to failures, maintenance, or upgrades of SD-WAN Controllers.

WAN Edge routers maintain these connections with SD-WAN Validators until either the failed SD-WAN Controller comes back online or is removed from the SD-WAN network by a network administrator. It is important to understand this behavior when you’re doing SD-WAN Validator capacity

planning. For example, in a balanced dual data center design (similar to the one shown on Figure 5-12), a data center outage would lead to every WAN Edge device losing a control connection to one of its “assigned” SD-WAN Controllers. This would result in all routers in the network establishing permanent, long-lived connections to SD-WAN Validators in the operational data center from each of their TLOCs. To handle such scenarios, it is crucial to provision enough SD-WAN Validators in each data center not only to handle all these long-lived connections but also to continue providing standard SD-WAN Validator services to devices attempting to join the network.

Note that SD-WAN Validators never participate in the data plane of the overlay network. Therefore, the failure of any SD-WAN Validator has no impact on data traffic.

SD-WAN Control Components Scalability

When designing a large Cisco SD-WAN network, it is important to be mindful of the capacity and scalability limitations of SD-WAN Control Components:

• **SD-WAN Manager:** Deployment options for SD-WAN Manager range from standalone nodes to 3- or 6-node clustered setups. A single instance can potentially handle up to 1000 to 1500 devices, while a 6-node SD-WAN Manager cluster may support up to 12,500 devices. However, the actual number of supported devices may vary significantly depending on factors

Technet24

such as SD-WAN Manager resources (instances/CPU/RAM/storage), the volume of generated statistics, and the version of the SD-WAN software. If the number of devices in the network exceeds the supported scale, another SD-WAN overlay may need to be considered.

• **SD-WAN Controller:** A single SD-WAN Controller can handle up to 5000 control connections and 2500 OMP sessions. The maximum number of tested SD-WAN Controller instances in a single Cisco SD-WAN overlay is 12.

While two SD-WAN Controllers usually offer sufficient redundancy in many deployments, larger networks might need more SD-WAN Controllers, along with the use of affinity features.

In a Multi-Region Fabric setup, SD-WAN Controllers can be configured to serve all regions. Alternatively, designated SD-WAN Controllers may be added and assigned to specific regions (both core and access), as needed.

• **SD-WAN Validator:** Each SD-WAN Validator supports up to 4000 connections, and up to eight SD-WAN Validators have been tested in a single Cisco SD-WAN domain. SD-WAN Validators can be oversubscribed under normal conditions as routers joining the network only make transient connections to SD-WAN Validator. However, additional SD-WAN Validators may be required to handle permanent long-lived connections from WAN Edge routers when they are out of equilibrium.

Note

All the scaling and sizing numbers discussed in this chapter are specific to Version 20.12 of the SD-WAN software. However, these numbers may vary depending on factors such as allocated resources, load, and the version of SD-WAN software in use. For more accurate specifications, please refer to the “Recommended Computing Resources for Cisco Catalyst SD-WAN Control Components” document for the SD-WAN software version you are using or are planning to use, available on the Cisco website.

Figure 5-13 summarizes these numbers.

Technet24

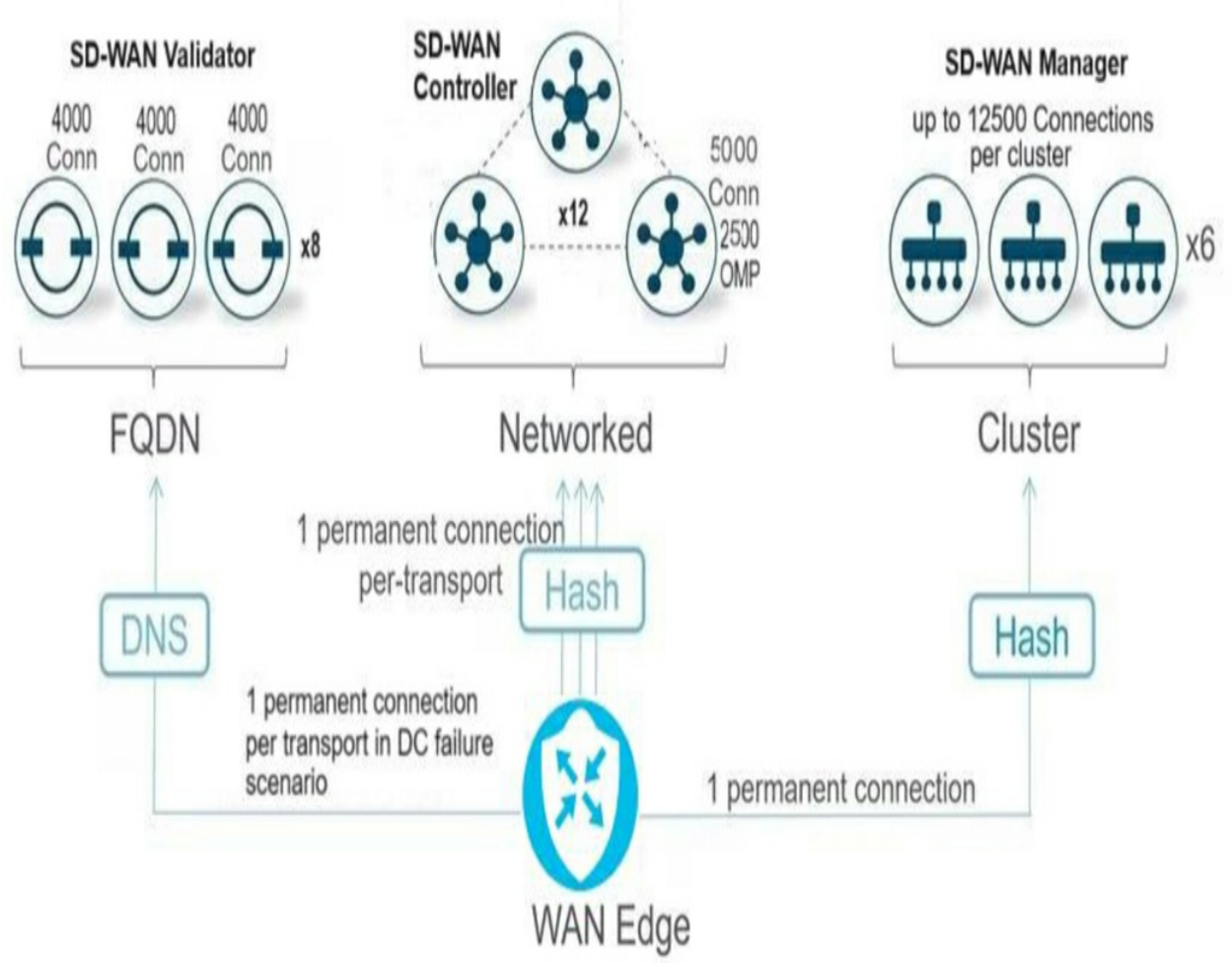


Figure 5-13 *Cisco SD-WAN Control Component Scalability*

Note that in a large-scale deployment, you may need to provide more resources to virtual machines running your SD-WAN Control Component instances. When designing an SD-WAN network, refer to the “Release Notes for Cisco Catalyst SD-WAN Control Components” and “Cisco Catalyst SD-WAN Control Components Compatibility Matrix and Recommended Computing Resources” documents, published on the Cisco website. These documents provide guidance on server resource requirements for SD-WAN Control Components, as well as the maximum number of devices supported

in a specific software release, along with other considerations.

SD-WAN Control Components Sizing Exercise

Let’s estimate an SD-WAN Control Components deployment for an imaginary network that has two data centers and 4000 sites in the following setup:

• 1000 small sites, served by one router with two transports (two TLOC routes per router)

• 1000 medium sites, with two routers with one MPLS and one Internet transport in each (two TLOCs per router)

• 1000 large sites, with two routers with one MPLS and one Internet transport in each (Transports are extended between routers for increased availability; therefore, each router is configured with four TLOCs.)

• SD-WAN Application Intelligence Engine (SAIE) enabled

• Cisco SD-WAN Release 20.12

Without data center routers, this network has 5000 WAN Edge routers and 13,000 TLOCs, as summarized in Table 5-2. Routers are using the defaults of two OMP sessions and two control connections per TLOC.

Table 5-2 *SD-WAN Control Components Sizing Example*

Technet24

Site Type	Number of Sites	Number of Routers per Site	Total Number of Routers	Number of TLOCs per Router	Number of TLOCs per Site	Total Number of TLOCs
Small	1000	1	1000	1	1	1000
Medium	1000	2	2000	2	4	4000
Large	1000	2	2000	4	8	8000
Total			5000			13,000

This network needs to be designed as fully redundant and should remain fully operational in the event of a data center outage. To meet these requirements, the following considerations apply to the SD-WAN Control Components:

• SD-WAN Manager should support 5000 devices. Therefore, based on the "Recommended Computing Resources for Cisco Catalyst SD-WAN Control Components Release 20.12.x" document, this design will require a 6-node SD-WAN Manager cluster for each data center.

• SD-WAN Controllers should support 10,000 OMP sessions (two per router) and 26,000 control connections (two per TLOC). That translates to 10,000/2500 = 4 SD-WAN Controllers to handle all OMP sessions and 26,000/5000 = 5.2 SD-WAN Controllers to handle all control connections. Rounding up, that is 6 SD-WAN Controllers in each data center.

• SD-WAN Validator orchestrators should support 13,000 long-lived control connections in a data center failure scenario (one per TLOC). That translates into 4 SD-WAN Validators per data center (rounding up from 13,000/4000 = 3.25).

Based on these calculations, SD-WAN Control Components for this network should be deployed as follows:

• **Data Center 1:** Active 6-node SD-WAN Manager cluster, six SD-WAN Controllers, and four SD-WAN Validator orchestrators

• **Data Center 2:** Standby 6-node SD-WAN Manager cluster, six SD-WAN Controllers, and four SD-WAN Validator orchestrators

This exercise is just a high-level example. In a real life, other considerations apply, such as sizing of the SD-WAN Control Components or statistics and route scale requirements. Therefore, for SD-WAN Control Components design in large networks, it is recommended to reach out to Cisco and validate the design before deployment.

Cisco SD-WAN Implementation Preparation

Once the design is finalized, the next crucial step for successful Cisco SD-WAN enablement is implementation planning. Thorough planning ensures success throughout all stages of SD-WAN implementation, including data center deployment, branch deployment, and policy configuration, as well as

Technet24

post-implementation day-to-day operations. The following are some of the most important preparations to make before starting the implementation:

• **Create a system IP address structure:** A system IP address, a mandatory configuration element in Cisco SD-WAN, is a system-level IPv4 address that uniquely identifies the device. It acts much like a router ID, and it is not advertised or known by the underlay. However, best practice recommends assigning this system IP address to a loopback interface and advertising it within one of the service VPNs. It can then be used as a source IP address for SNMP and logging, making it easier to correlate network events with SD-WAN Manager information.

It is recommended to develop a logical scheme for system IP addresses to make sites more easily recognizable.

• **Develop site ID scheme:** A site ID, another mandatory configuration element in Cisco SD-WAN, serves as a unique identifier of a site within the SD-WAN overlay network, with a numeric range from 1 to 4294967295. The site ID must be the same for all the Cisco SD-WAN devices that reside at the same site, where a “site” is a physical or virtual location within an organization’s network infrastructure, like a branch office, data center, or cloud instance.

A site ID scheme should be chosen carefully to streamline the policy application. Centralized policies are applied to blocks of sites, defined as a

list or range of site IDs (for example, 100,200–299), without wildcard support. To make an SD-WAN network more scalable, it is recommended to use ranges to group together sites with similar policies. This way, adding a new site with a site ID that falls into the matching range of policy will automatically cause the policy to be applied to it.

Table 5-3 provides an example of a six-digit site ID scheme suitable for a global organization with two data centers and two types of sites. In this example, policies specific to data centers in US West are applied to the site ID range 110000–110999. Policies specific to Type A sites in Canada East are applied to the site ID range 121000–121999. Policies specific to all Type B sites in all locations are applied to the site ID list that includes all the regional Type B sites ranges: [112000–112999, 122000–122999, 202000–202999, 302000–302999].

Table 5-3 *Cisco SD-WAN Site ID Scheme Example*

Technet24

Digit	Representation	Values
First	Continent	1=North America, 2=Europe, 3=Africa
Second	Region	0=unused, 1=US/Canada West, 2=US/Canada East
Third	Site type	0=Data Center/Hub, 1=Site Type A, 2=Site Type B
Fourth through sixth	Site/store number	001, 002, 003, etc.

The example sites are grouped together based on their geography, which is helpful in cases where you might want to prefer a regional data center over another. These are some of the other potential grouping criteria:

• Sites that access the Internet via a centrally located firewall versus those using Direct Internet Access

• Sites with low-end routers and sites with more powerful devices (Low-end routes support only a handful of SD-WAN tunnels and require a hub-and-spoke topology, while sites with higher-end routers may use a full-mesh topology.)

• Different SLA and transport requirements, such as certain sites using MPLS for all the traffic, with the Internet circuit serving only as backup, and some sites using MPLS for voice only while routing everything else through the

Internet circuit

You can have overlapping types, but the general idea is to put them in categories that makes it easier to apply policy from a configuration perspective. It is strongly recommended to review the requirements and required policies before assigning site IDs.

• **Design Cisco SD-WAN policy:** Cisco SD-WAN policy dictates the groups of interest (prefix lists, site lists, and so on), network topology, traffic engineering, overlay routing, application-aware routing, security posture, quality of service classification, application visibility, and more. Prior to implementation, the Cisco SD-WAN policy should be fully thought out so that sites rolling onto the fabric inherit the policy and adhere to it immediately. A well-constructed and well-executed policy will ensure that the implementation is smooth and that post-implementation performance is high.

• **Collect physical and logical diagrams for data center and branch sites:** To have a good understanding of the existing architecture at data centers and branch sites, it is important to collect up-to-date physical and logical diagrams. This way, you can ensure that there is sufficient power, cooling, rack space, and port density, and you can visualize how the WAN edge routers will be connected to the existing network.

• **Collect existing device configurations:** Collecting and reviewing the

Technet24

existing device configurations ensures that all considerations have been made around Cisco SD-WAN feature support and enablement. You don’t want to realize, in the middle of a site turnup, that a specific feature you were using prior to the Cisco SD-WAN migration is not yet supported or that you have not made provisions to ensure that the feature would still function as designed after the migration.

• **Analyze existing topology, routing, and traffic engineering:** Having a good understanding of existing routing and traffic engineering in the network will allow you insert Cisco SD-WAN into the network with ease. It will allow you to determine if anything needs to be changed during the migration or if a specific consideration needs to be made to ensure that the same network behaviors exist post-migration. This might include topology design, data center affinity configuration, equal-cost multipath routing, high availability configuration, and failover behavior.

• **Allocate new physical and logical network resources:** If there are new WAN Edge routers added to the network (for example, in data centers), physical connectivity will need to be provided for the LAN and WAN. New subnets and IP addresses will need to be allocated for transit and management networking. Planning this ahead of time will allow you to scale consistently and contiguously.

• **Prepare integration with the existing infrastructure:** It is important to plan the integration process to ensure interoperability and a smooth transition

from current setups to the SD-WAN environment.

• **Plan application performance management:** You need to classify applications in the network (for example, real-time, business-critical, standard, or best-effort) and understand their SLAs regarding packet loss, latency, and jitter tolerance. Consider whether applications have a preferred transport (for example, MPLS for voice) and plan for the scenarios where the preferred transport is unavailable or SLAs are not met.

• **Review QoS requirements:** If necessary, establish QoS policies across the SD-WAN to ensure that critical applications receive the bandwidth and priority they require.

Cisco SD-WAN Transport Connectivity

As detailed in previous chapters, a WAN Edge router has a transport-side VPN and a service-side VPN that need to be integrated into the existing network. For the transport side, Cisco highly recommends that all WAN Edge routers have connectivity into all available transports at a site in order for high availability and for application-aware routing to work optimally.

Transport Connectivity: Single-Router Sites

The most common way to integrate transport-side connections on a WAN Edge router into the network is by dedicating a single interface in VPN 0 for each carrier/transport. Routing can be as simple as a default route to the next-

Technet24

hop carrier edge equipment or gateway. The only requirement is that the IP address of the WAN Edge router transport interface must be reachable on that particular underlay.

One way to leverage an existing MPLS transport as a part of the Cisco SD-WAN overlay could be by connecting a VPN 0 interface directly to the MPLS CE router over a /30 transit network. A static default route toward that MPLS CE router is configured, and the MPLS CE router must advertise that transit network into the underlay.

Leveraging an Internet transport is nearly an identical process, except that the VPN 0 interface can connect to an existing DMZ or Internet edge firewall. It can be provided either a public or private IP address on a larger subnet, and it might or might not be behind NAT. Figure 5-14 details this type of connectivity.

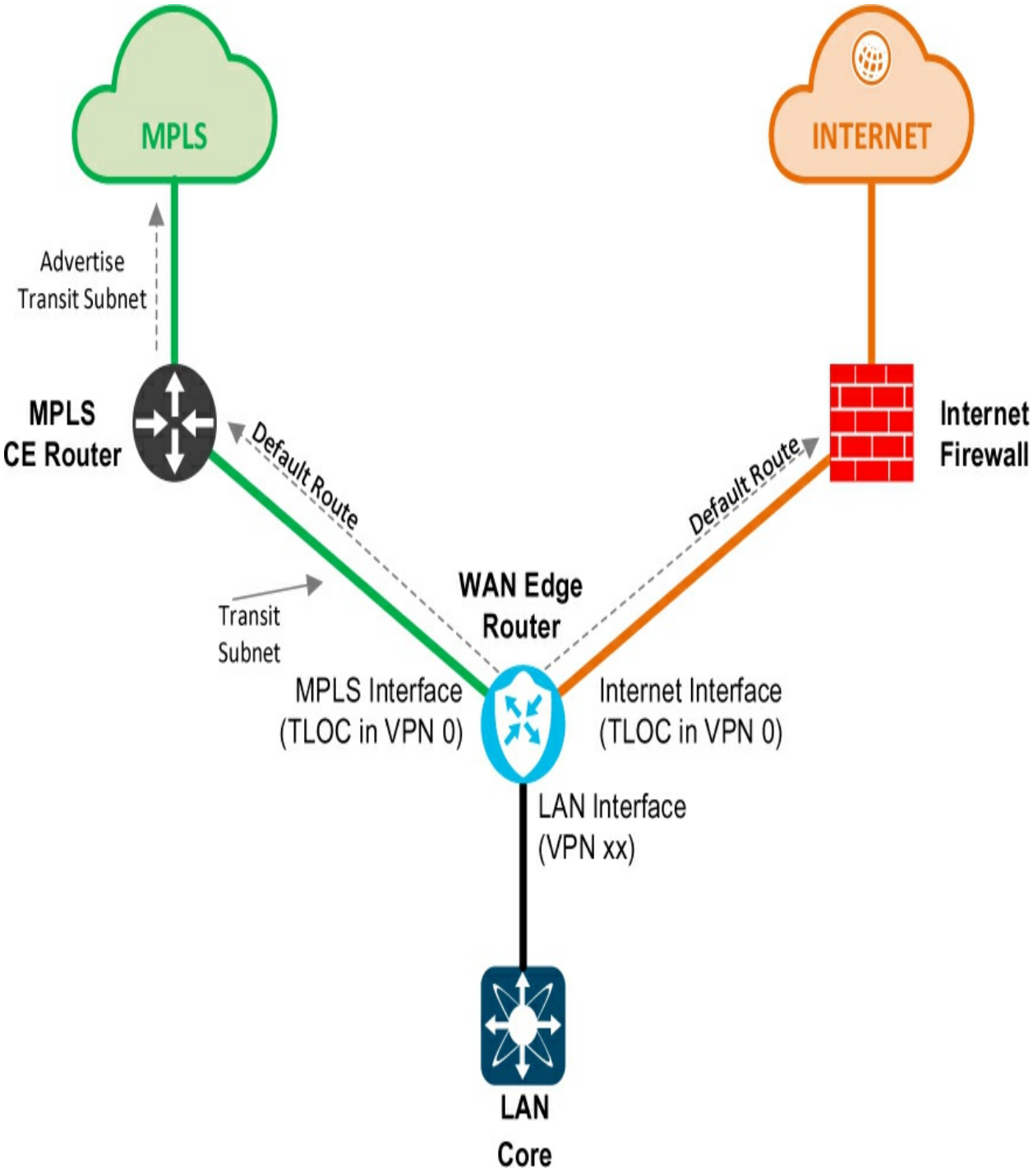


Figure 5-14 *Direct Transport Connectivity*

Note

If NAT is required in order to provide connectivity to the

Technet24

Internet for the WAN edge router, it is recommended to configure static one-to-one NAT whenever possible to avoid tunnel bring-up issues that might occur in some scenarios.

Indirect switched connectivity to transports is also an option and can be accomplished through the use of VLANs and dedicated interfaces or subinterfaces. In the example shown in Figure 5-15, the MPLS CE router and Internet firewall are physically connected to L2 ports on the LAN core, where three VLANs are provisioned: VLAN 10 for the LAN service-side connection, VLAN 20 for the MPLS transport connection, and VLAN 30 for the Internet transport connection. On the WAN Edge router, the subinterfaces for the MPLS and Internet transport connections share the same physical interface with the service-side link to the LAN core instead of using separate dedicated interfaces, minimizing the number of interfaces used. Routing to the MPLS CE router and Internet edge firewall remains the same as in the directly connected design.

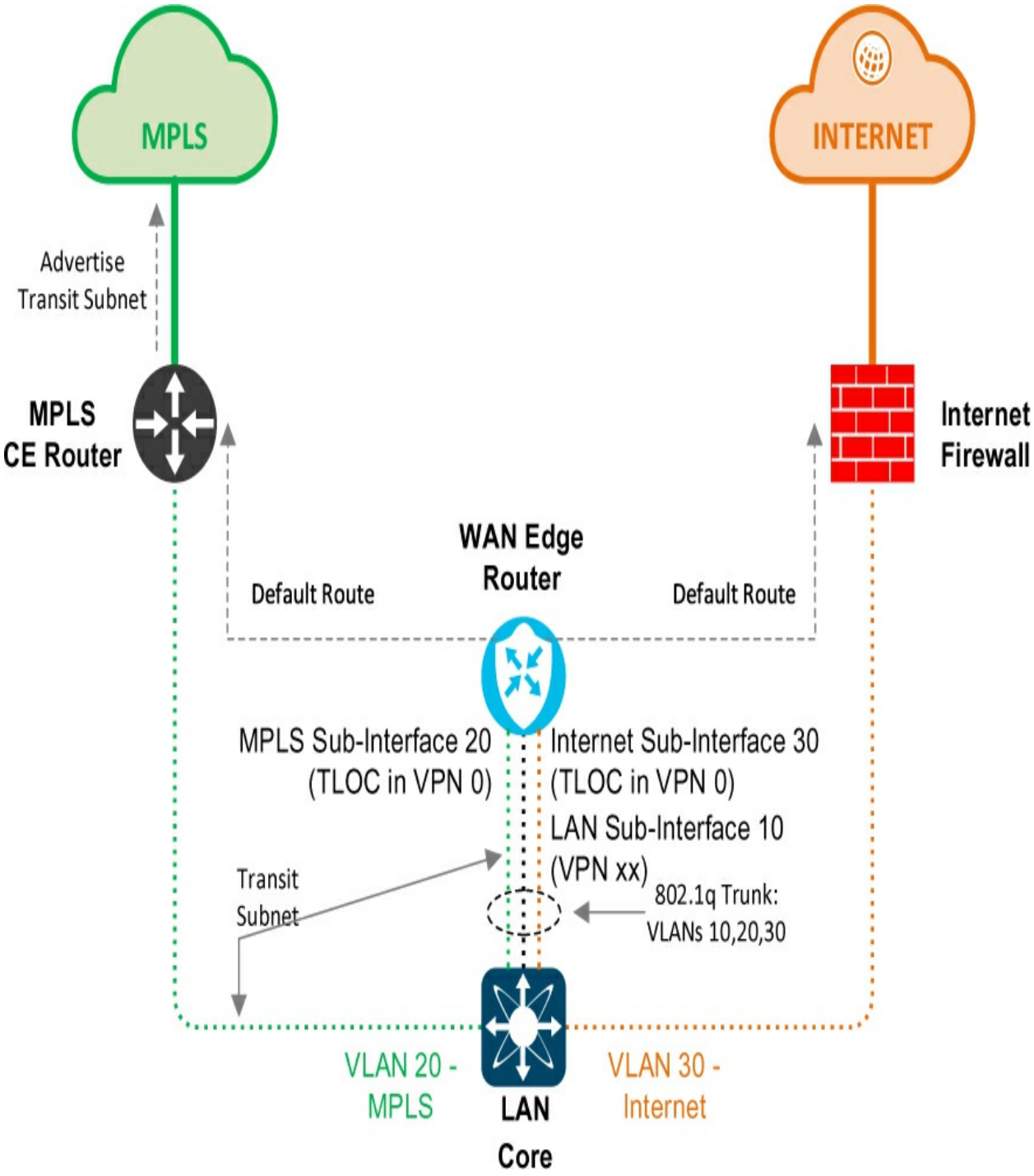


Figure 5-15 *Indirect Switched Transport-Side Connectivity*

In certain scenarios, a transport may not be exclusively dedicated to SD-WAN. For example, this could occur when Internet access at the data center

Technet24

is centralized and shared by all applications, including SD-WAN, or during a migration to Cisco SD-WAN when existing MPLS transport is used by both SD-WAN and legacy networks, and the MPLS CE router cannot be reconfigured to accommodate SD-WAN. In such cases, indirect routed connectivity may be used, where a WAN Edge device routes via another device(s), typically in the LAN core, to reach the appropriate transport, as illustrated in Figure 5-16.

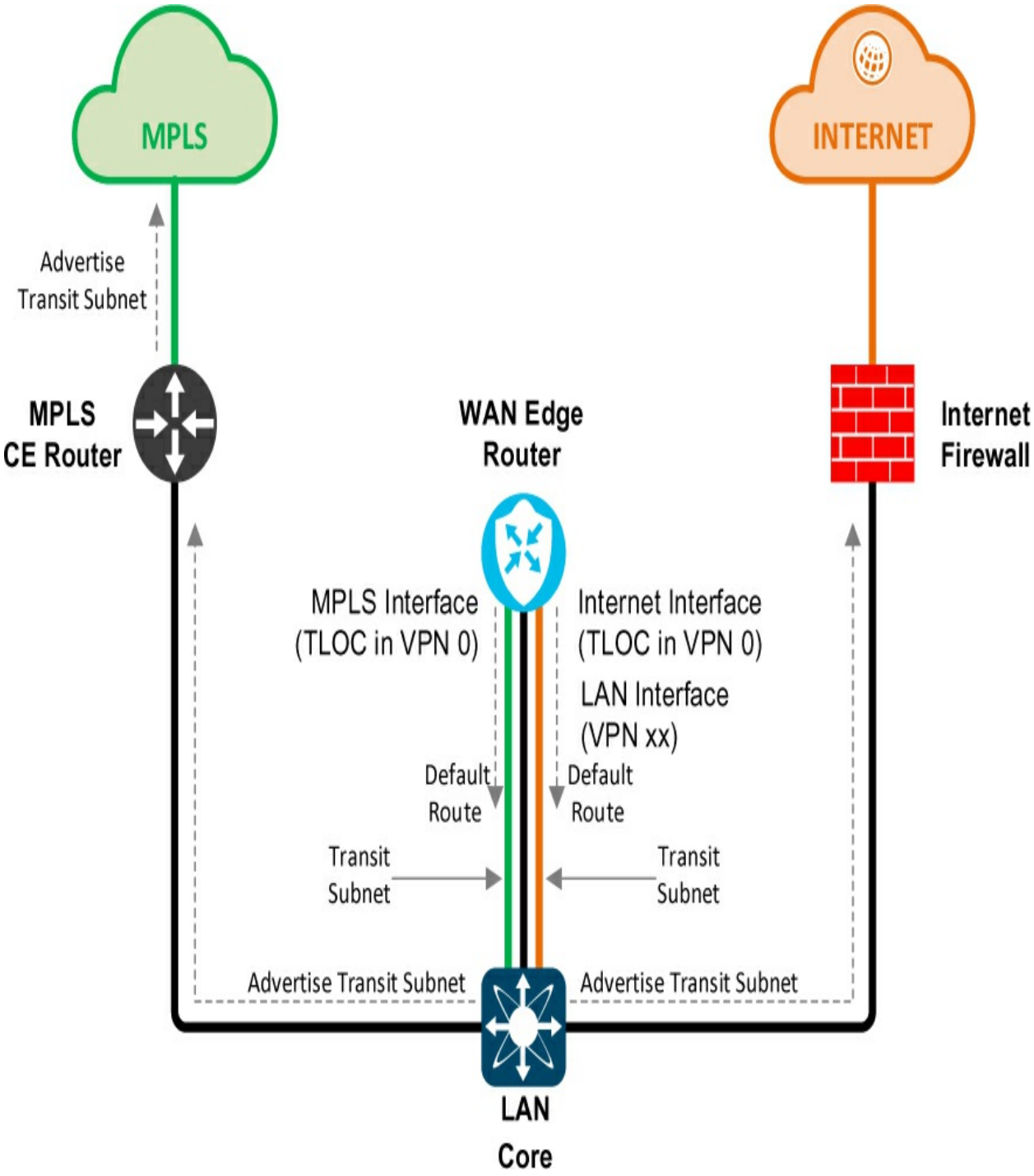


Figure 5-16 *Indirect Routed Transport-Side Connectivity*

However, this approach requires meticulous planning to prevent routing loops and route leaking between the SD-WAN underlay and overlay.

Technet24

Furthermore, there might be multiple redundant paths in the LAN core to reach the MPLS CE router or Internet firewall. Consequently, when Cisco SD-WAN measures path characteristics and evaluates performance, the results may be unpredictable, potentially impacting the efficiency of SD-WAN traffic handling. Therefore, this approach is the least preferred, and, if implemented, should be planned very carefully.

When implementing indirect routed transport connectivity, or in some other cases where underlay path manipulation or traffic engineering is required, enabling dynamic routing on a VPN 0 transport interface may be necessary. In such scenarios, a routing protocol such as BGP or OSPF can be configured in order for the router to learn more specific prefixes, and traditional route policies can be applied inbound and outbound.

Note

When using a routing protocol on VPN 0 to gain access to the underlay, it is important to ensure that the correct and optimal path to peer tunnel endpoints for that particular transport is followed through the local network.

Transport Connectivity: Dual-Routers Sites

When deploying two WAN Edge routers at a site, one approach is to reuse single-router designs.

Figure 5-17 illustrates how direct transport connectivity may be implemented in a dual WAN edge deployment, providing two options.

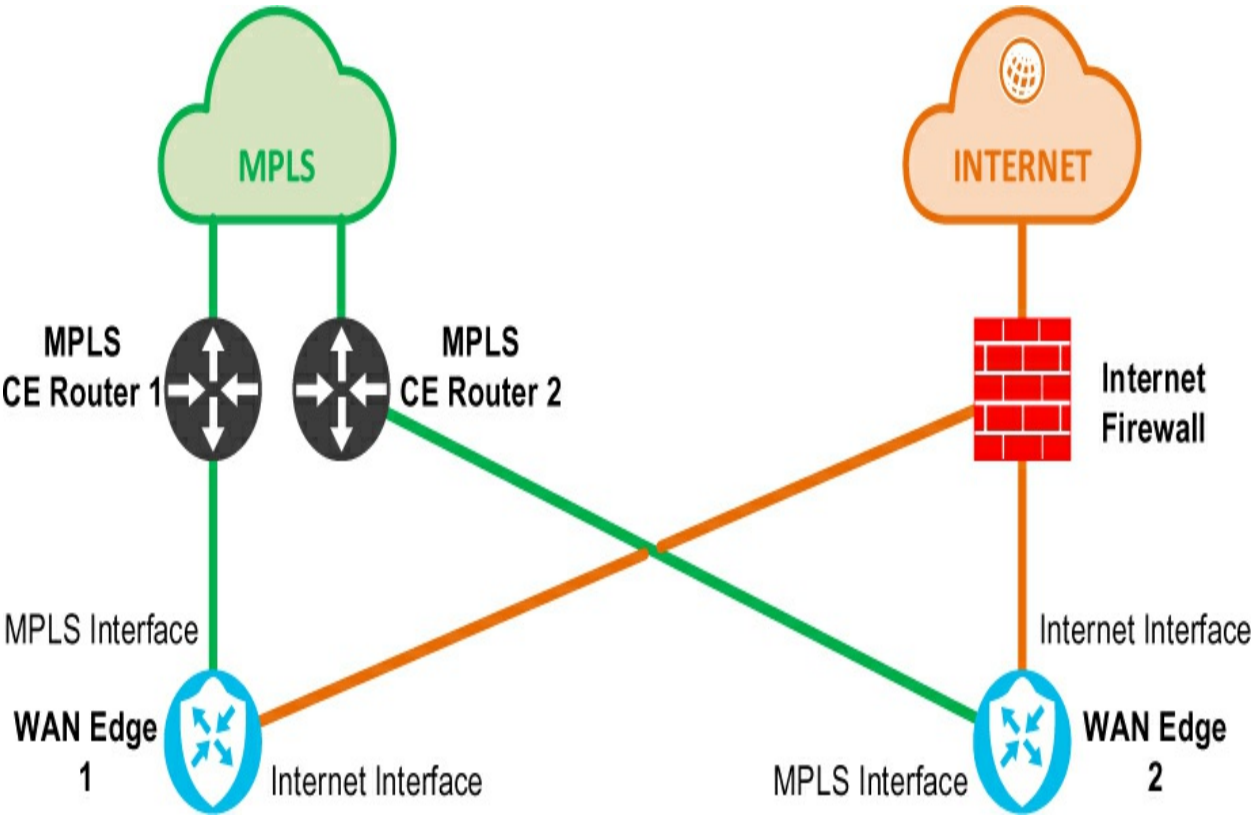


Figure 5-17 *Direct Transport-Side Connectivity with Dual Routers*

The first option, shown for MPLS transport, is to provision an additional MPLS CE router, with both WAN Edge routers directly connected to their own independent MPLS transports. This approach is suitable for environments where high availability is paramount, such as in the medical or financial sectors. However, it is generally not preferrable due to increased costs.

For Internet transport, an additional port is provisioned on the Internet firewall. While simpler to implement, this method still requires an extra port

Technet24

on the firewall and additional cabling work. If this presents a challenge, an alternative is to use indirect switched connectivity, as illustrated in Figure 5-18.

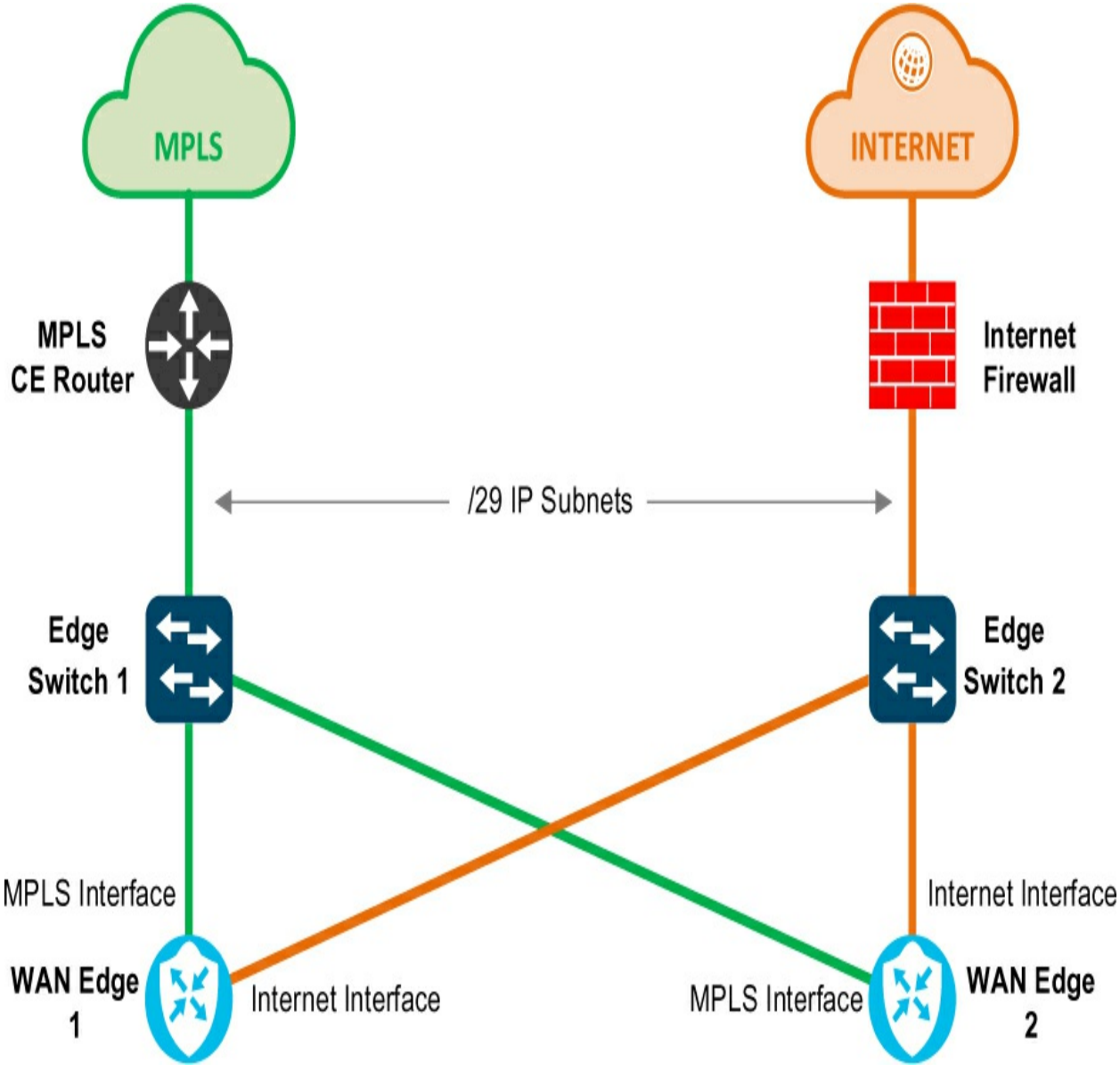


Figure 5-18 *Indirect Switched Transport-Side Connectivity with Dual Routers*

In this setup, interfaces of the transport gateway (the MPLS CE router or

Internet firewall) and both WAN Edge routers are in the same VLAN and IP subnet. This configuration may be acceptable if switching infrastructure is already available but may require additional investment if not.

The bigger challenge is the need to increase the size of the transit network to /29 instead of /30 to accommodate all three devices in the same IP subnet; this needs to be negotiated with the MPLS provider.



TLOC Extensions

It is a very common scenario to have only a single physical handoff and a single IP address on a given transport and yet have that transport terminated into two WAN Edge routers. In this scenario, public color and private color TLOC extensions can be used to meet the requirement.

A *TLOC extension* is a direct or indirect connection between two WAN Edge routers that extends a locally terminated TLOC to a peer router. This allows a WAN Edge router to leverage a non-locally connected transport as a part of the Cisco SD-WAN overlay. Figure 5-19 shows TLOC extensions configured between two WAN Edge routers.

Technet24

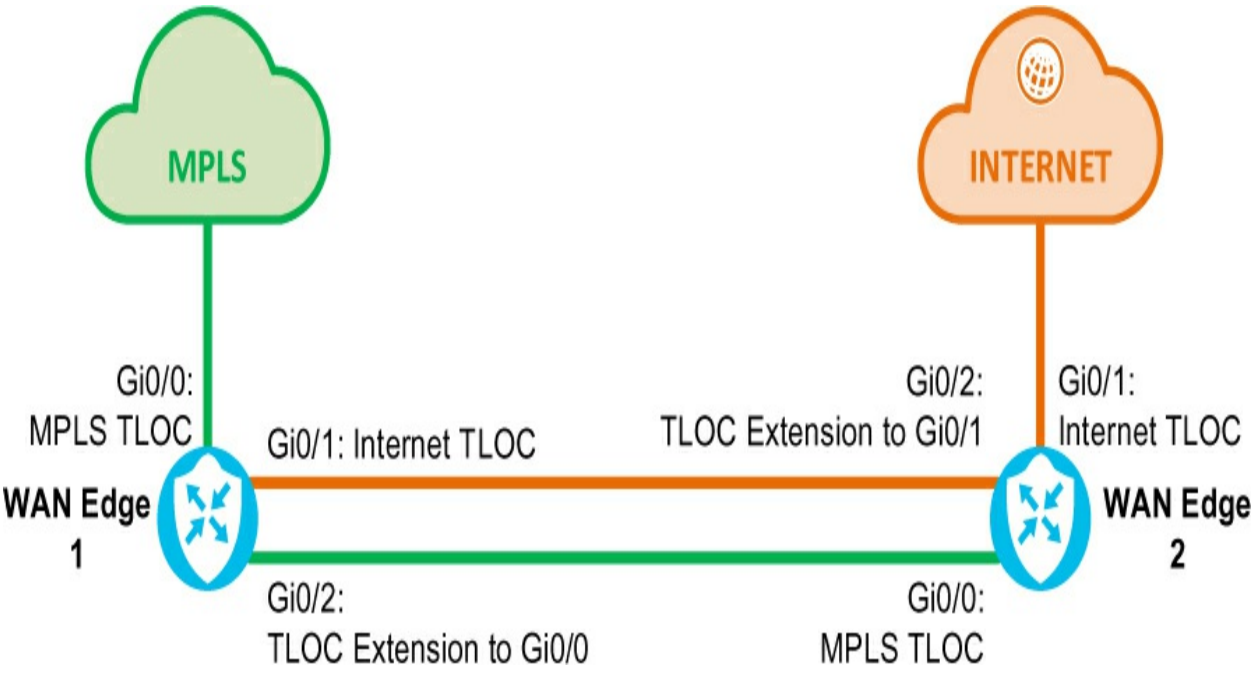


Figure 5-19 *TLOC Extension Dedicated Interfaces*

Multiple TLOCs can be extended from one WAN Edge router to another, but each TLOC extension requires its own dedicated interface on each router. In scenarios where there are not enough physical interfaces available, it is acceptable, if bandwidth allows, to use subinterfaces on a physical interface to configure TLOC extensions. This can be done either directly between two routers or through a LAN core.

In addition, WAN Edge routers support Layer 3 TLOC extension via GRE. This removes the requirement for the routers to have direct Layer 2 adjacency to one another. Hence, they can be geographically separated (as might be the case in a campus environment with multiple data centers).

When extending a private color via TLOC extension, you need to allocate a new enterprise routable transit network for the connection between WAN

Edge routers and ensure that it is reachable from the service provider's network. One way to achieve this is to ask the service provider to configure all the necessary routes statically. Alternatively, you can configure a dynamic routing protocol in VPN 0 on the router extending the locally connected private transport, peering with the service provider to advertise reachability of the new transit network. A route policy can be configured to filter inbound and outbound routes so that only the transit network is advertised out, and no other networks are learned. Remember that in most cases, a default route toward the service provider is all that is needed for tunnel building and control plane connectivity.

The WAN Edge router receiving the extension is not aware of anything different and configures its end of the extension as a standard private TLOC with a default route pointing toward the adjacent next-hop IP address on the newly created transit network. Figure 5-20 highlights the private TLOC extension.

Technet24

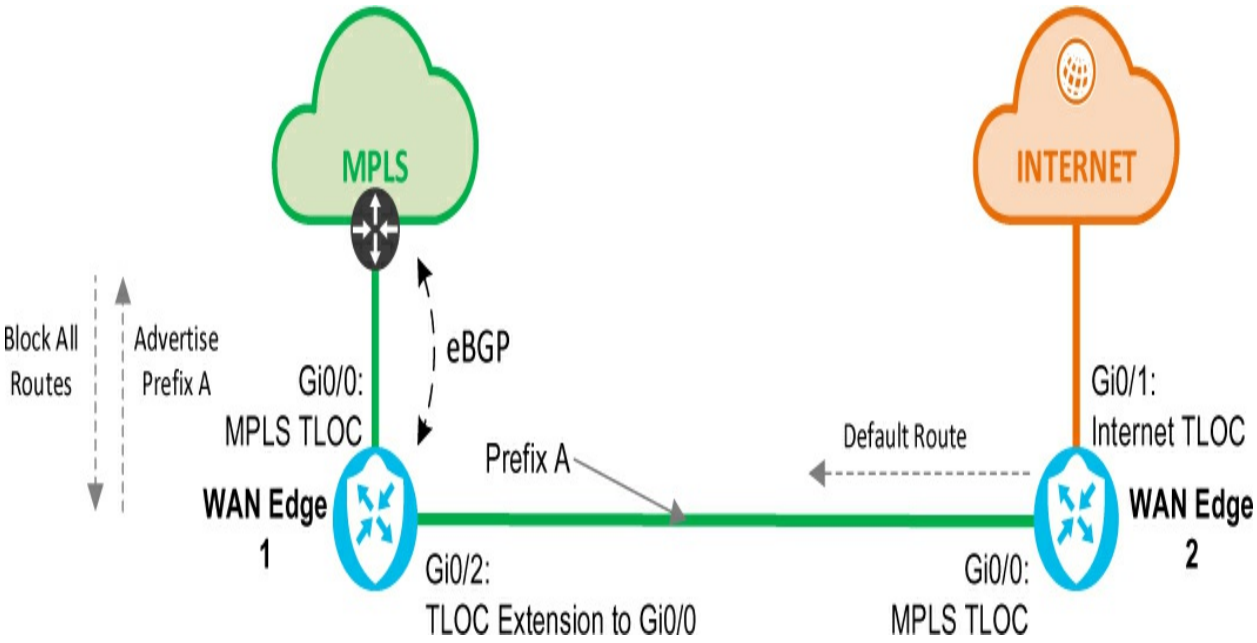


Figure 5-20 *Private TLOC Extension*

Note

It is important to ensure that the WAN Edge router receiving the extension is able to build control connections through the peer by allowing the transit network to be learned by the data center (if the SD-WAN Control Components reside there) or is allowed through the Internet edge firewall, where it can egress to reach the cloud SD-WAN Control Components.

Extending a public color via TLOC extension is configured a bit differently than extending private TLOC extensions. For a public TLOC extension, it is necessary to enable NAT on the egress interface toward the transport on the router extending the public transport, if it’s not already enabled. NAT facilitates private-to-public mapping, allowing WAN Edge routers to

communicate with other remote sites and SD-WAN Control Components without needing to advertise the private transit network. This enables the allocation of any locally significant private transit network between WAN Edge routers to build connectivity.

As with a private TLOC extension, the WAN Edge router receiving the extension is not aware of anything different and configures its end of the extension as a standard public TLOC with a default route pointing toward the adjacent peer's next-hop IP address on the newly created transit network. Figure 5-21 shows an example of a public TLOC extension.

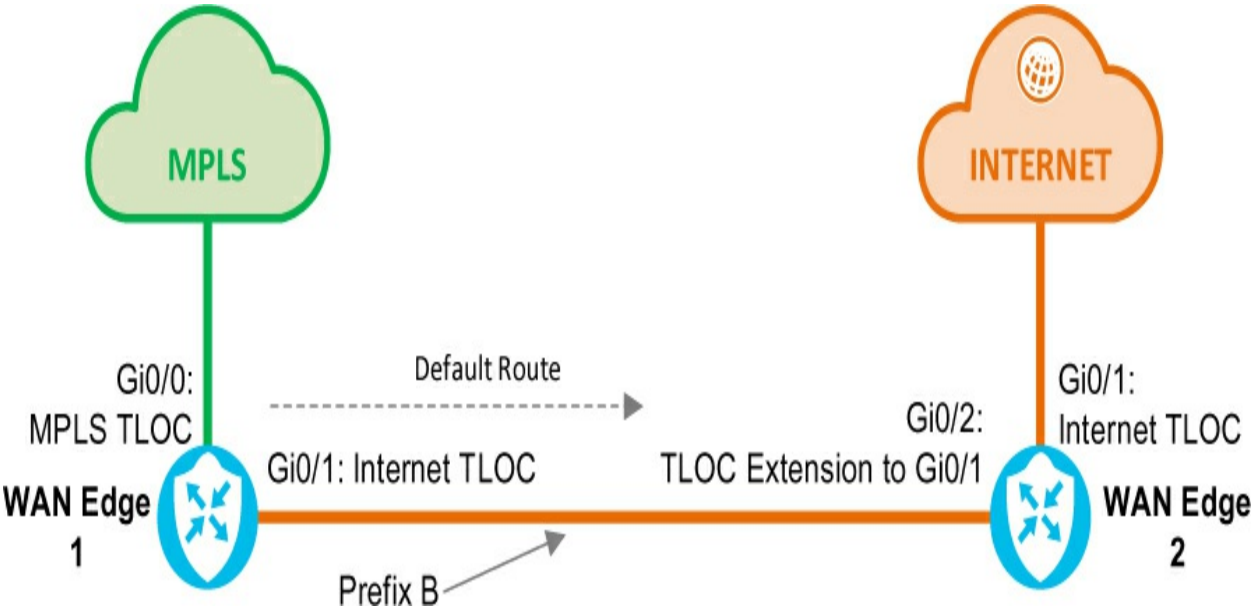


Figure 5-21 *Public TLOC Extension*

Note

Depending on your branch design, it is not always necessary to mutually extend TLOCs between WAN Edge routers. In an

Technet24

active/backup branch setup, extending a transport from the primary to the secondary WAN Edge router is redundant because the secondary backup router only becomes active when the primary goes offline. Therefore, when the secondary WAN Edge router is active, a path via the primary router is unavailable and cannot be used anyway. To implement this design, it is simpler to terminate the MPLS circuit on the primary WAN edge router and the Internet circuit on the secondary WAN Edge router and extend it to the primary, as shown in Figure 5-21. This approach eliminates the need to manage routing for the transit network between routers.

In some advanced topologies, an organization may want to leverage multiple CE routers and circuits to the same service provider for overlay connectivity and may also want to retain granular path selection and control via Cisco SD-WAN policy. Because a single transport color cannot be used twice by the same WAN Edge router, two alternative TLOC designs involving Loopback interfaces exist to meet these requirements.



Loopback TLOC Design

A loopback interface can be configured as a conventional TLOC and be advertised toward both CE routers via two connections (one to each CE).

This way, both CE routers can reach the Cisco SD-WAN tunnel endpoint configured on the loopback, and the WAN Edge router can reach the underlay via both CE routers. The drawback to this solution is that it requires additional routing complexity. In addition, because there is only a single color configured, path selection granularity via Cisco SD-WAN policy is also limited.

Two modes exist for this loopback design: bind mode and the unbind mode. In bind mode, the loopback interface is bound to a physical interface, and traffic destined to the loopback interface will be carried to and from the mapped physical interface. This design can be used when you have connected subnets on the transport side and want to use a loopback to form control connections and data tunnels. Figure 5-22 shows the bind mode option.

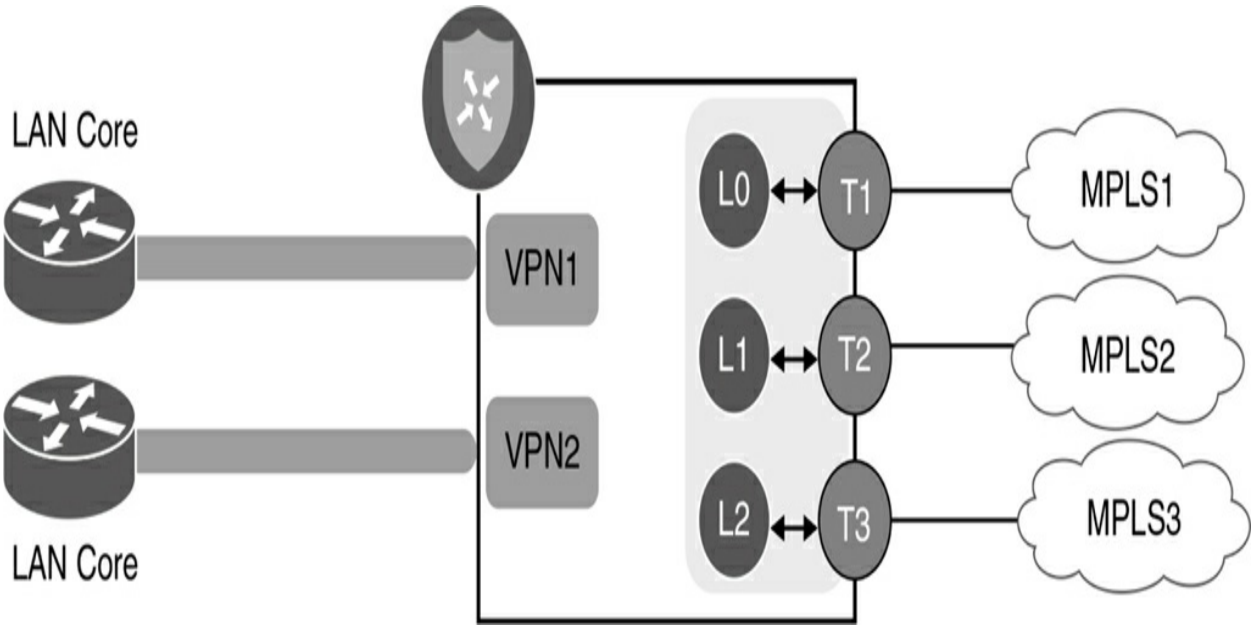


Figure 5-22 *Loopback TLOC Bind Mode*

Technet24

In the unbind mode, the loopback interface is not bound to any physical interface. Traffic destined to the loopback interface can go through any physical interface based on a hash lookup. Figure 5-23 illustrates this.

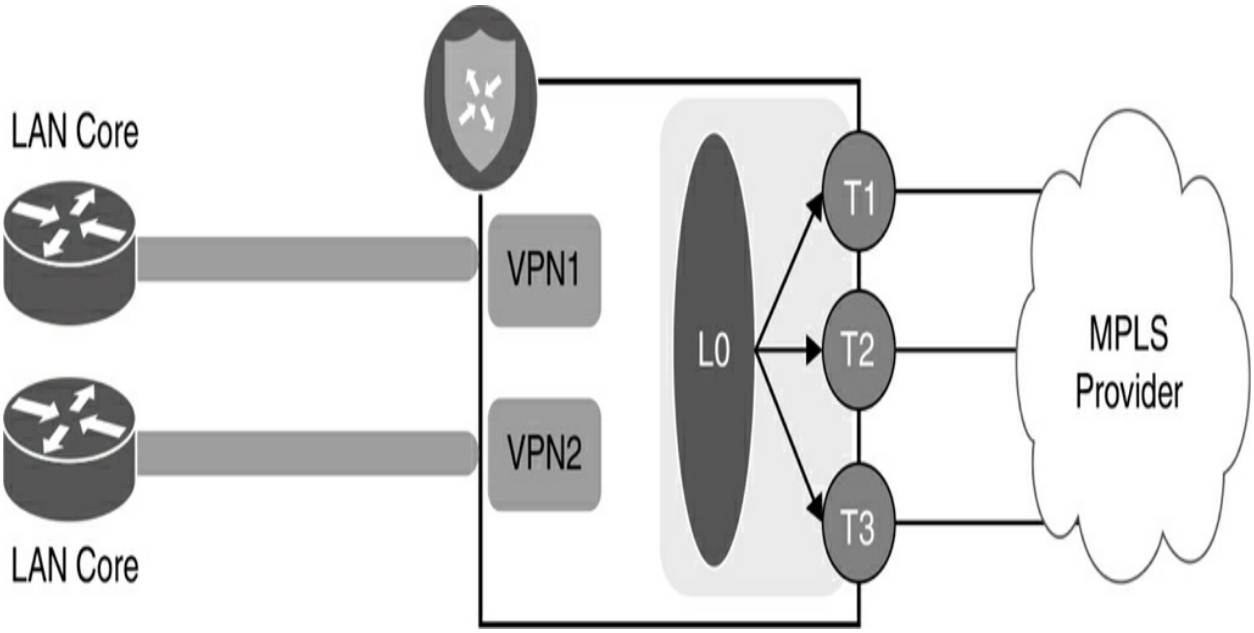


Figure 5-23 *Loopback TLOC Unbind Mode*

Note

It is important to note that in the unbind mode, information collected via Bidirectional Forwarding Detection (BFD) can be on any of the interfaces. Because of this, the liveliness info can be seen as different for each remote site, even though the remote sites are on the same local color at this location.

Another scenario where loopback interfaces prove beneficial is illustrated in Figure 5-24. This situation typically arises with the Internet, when you have

multiple transport options in the same underlay at remote branches and a single transport at the hub or data center. With the current Cisco SD-WAN functionality, applying data or application-aware routing policies to prefer a specific transport from branch toward hub grants full control over traffic flow at the branch side. However, at the hub, where only one local transport is available during policy configuration, controlling which path to take to the remote site and which remote transport receives traffic from the hub becomes challenging.

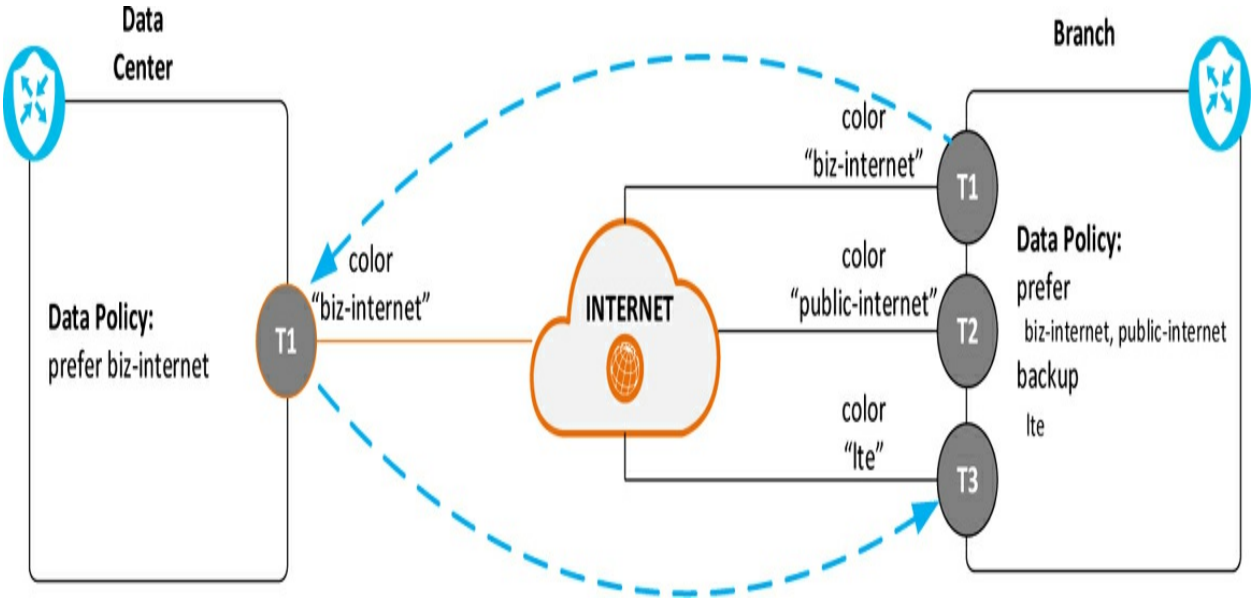


Figure 5-24 *Multiple Remote Colors*

In this example, there are three data tunnels present at the hub site WAN Edge router:

• Local **biz-internet** color to remote **biz-internet** color

• Local **biz-internet** color to remote **public-internet** color

Technet24

• Local **biz-internet** color to remote **lte** color

When the policy prefers the local biz-internet color, all three tunnels match it, and traffic is load-balanced among them, potentially landing on any remote transport, including LTE, which, in this case, is costly and supposed to be used only as a backup.

The workaround is to create loopback interfaces in bind mode at the hub for remote branch colors that are not physically present there. This eliminates ambiguity during policy configuration, allowing for a desired level of policy granularity at hub sites, as shown in Figure 5-25.

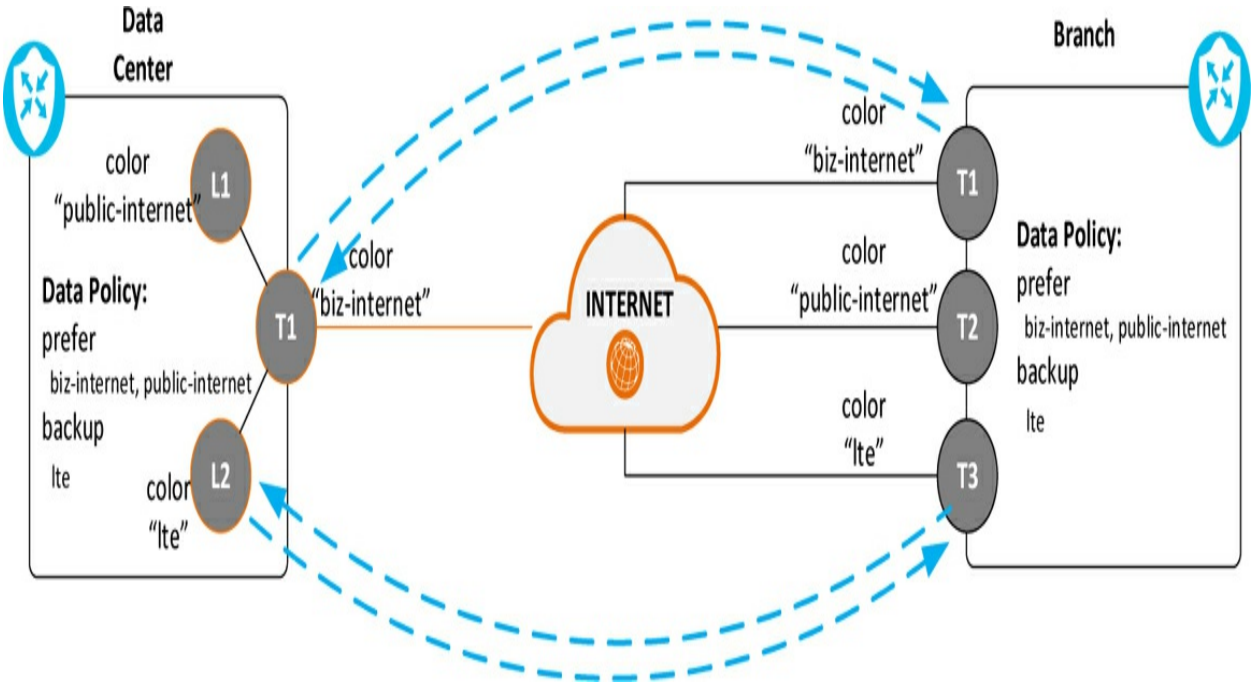


Figure 5-25 *Multiple Remote Colors with Loopback TLOCs*

In this case, there are three data tunnels present at the hub site WAN Edge router:

• Local **biz-internet** color to remote **biz-internet** color

• Local **public-internet** color to remote **public-internet** color

• Local **lte** color to remote **lte** color

Now, when policy prefers the local biz-internet color, traffic is only sent via the first tunnel, deterministically landing only on the biz-internet remote transport. Consequently, LTE is used solely as a backup, minimizing unnecessary usage charges. Moreover, this approach enables the configuration of transport parameters on the data center side, such as enabling the low-bandwidth-link feature for LTE, which further optimizes LTE bandwidth utilization.

Cisco SD-WAN Data Center Design

In nearly all cases, the data center is the first site to be implemented in Cisco SD-WAN deployments. While there are many ways to integrate SD-WAN into the data center, the most common and strongly recommended approach is to implement the solution in parallel with the existing WAN infrastructure. This is accomplished by standing up WAN Edge routers alongside the current WAN setup (behind the MPLS CE/PE routers and Internet edge firewalls) and providing the routers with indirect connectivity to WAN transports. This approach is particularly favored when an organization doesn’t have the luxury of providing dedicated circuits for WAN Edge routers or when handoffs for all the transports are already in service.

Technet24

It is common to see new WAN Edge devices leveraging private connectivity through an existing MPLS CE/PE router and public connectivity through an existing secure Internet edge firewall. Figure 5-26 illustrates this architecture.

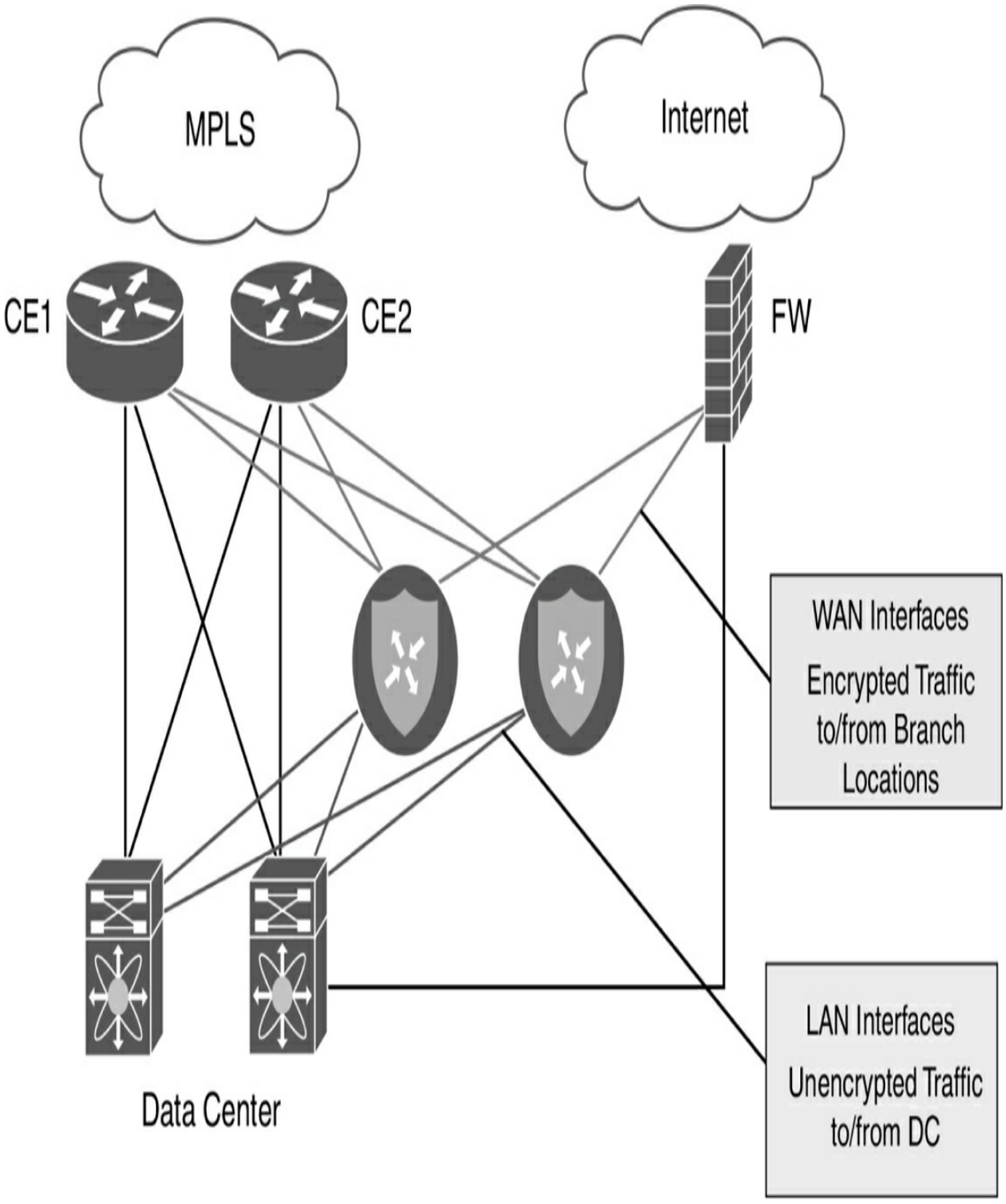


Figure 5-26 *Data Center Dedicated Pod*

Note

Technet24

WAN Edge routers come with built-in security features and do not need to sit behind firewalls. In branches, WAN edge routers often connect directly to the Internet transports. However, data centers typically have established Internet access segments protected by firewalls, which WAN Edge routers commonly leverage, as illustrated throughout this section.

Nevertheless, some organizations opt for dedicated Internet access within data centers solely for SD-WAN. In such cases, they procure new Internet circuits specifically for SD-WAN and often connect them directly to routers without firewalls in front of them. This approach is perfectly valid, as long as it aligns with the security policies of the organization.

This design allows the data center to act as a transit site for traffic between Cisco SD-WAN and non-SD-WAN sites and allows for a gradual migration of remote sites to Cisco SD-WAN without affecting the legacy network. This type of design also decouples and removes fate sharing between non-migrated and migrated sites, as they are landing on separate routers in the data center with distinct domains for routing and control. This same design can be used when migrating from legacy overlay technologies, such as Dynamic Multipoint VPN (DMVPN) and Group Encrypted Transport VPN (GETVPN), to Cisco SD-WAN.

In some environments, transiting the hub and the potential for added latency may be a concern. Designating multiple regional hubs as transit points for dedicated geographies and configuring the overlay routing to leverage these hubs intelligently is a common solution to this problem. For example, you may want to designate a U.S. West hub, a U.S. East hub, an APAC hub, and an EMEA hub, with each of these hubs handling the transit traffic between its respective regions.

An alternative, albeit much more complex and operationally expensive, method of solving the same latency problem is to mix overlay and underlay networking to provide a direct path between non-migrated and migrated sites. The section "Integrating Cisco SD-WAN with Existing Networks" addresses this approach later in the chapter.

Transport-Side Connectivity

Data centers form the heart of a network, offering extensive infrastructure and connectivity options. Typically, WAN Edge routers are deployed in data centers in pairs in a dual-router setup, with direct or indirect switched connectivity to transports, as previously discussed. TLOC extensions may also be used when transports are physically connected to WAN Edge routers and need to be accessible to other routers.

Service-Side Connectivity

Service-side connectivity is commonly achieved by provisioning one or two

Technet24

interfaces in a service VPN and connecting them into the data center core. The data center core is typically a point in the network where aggregation of legacy routes and Cisco SD-WAN routes occurs; it allows for natural transitive connectivity. Usually, a routing protocol (such as BGP) is run between the WAN Edge router and the data center core so that the WAN Edge router can learn both data center and legacy WAN routes as well as advertise migrated site routes back toward the data center.

External BGP (eBGP) is typically the protocol of choice, as it provides built-in loop prevention mechanisms (via AS_Path and Site of Origin attributes) and has a very flexible and comprehensive path selection algorithm. This algorithm can be manipulated in both directions via route policy. For instance, a route policy could match a set of routes from the overlay and set a local preference in order to influence the data center core routing. BGP Multi-Exit Discriminator (MED) attribute can also be leveraged for path selection and is actually carried within OMP for SD-WAN Controller path selection in the overlay. Finally, BGP provides granular tagging and filtering mechanisms through the use of route maps, prefix lists, and communities. Figure 5-27 shows this type of connectivity.

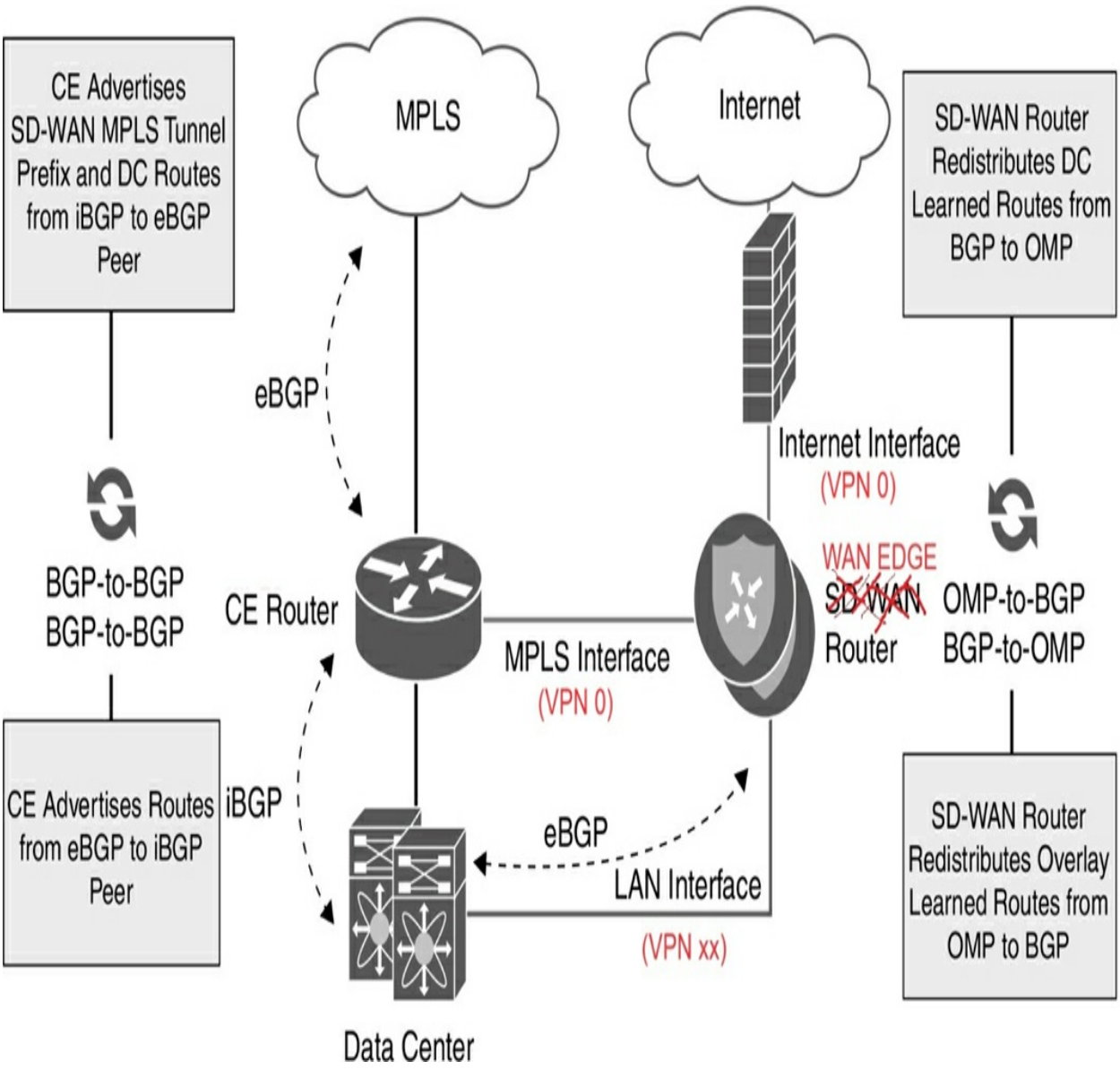


Figure 5-27 *Service-Side BGP*

While eBGP is usually the recommended service-side routing protocol, depending on the platform, Cisco SD-WAN supports iBGP, OSPF, EIGRP, RIPv2, static routing, and configuration of the first-hop redundancy protocol VRRP as well. Selecting an appropriate service-side routing protocol for an organization depends completely on the existing network configuration, required features and functionality, and operator comfort level.

Technet24

In some instances, using a service-side routing protocol that is different from the existing routing protocol deployed in the data center core can add complexity—especially when it comes to route redistribution between the two protocols. To avoid this complexity, it may be worth considering the same routing protocol on the WAN Edge router that is found on the data center core.

An alternative approach is to connect and peer the service side of the WAN Edge router directly with an existing MPLS CE router that is already running BGP. This way, no additional redistribution needs to occur on the data center core since the MPLS CE router is already providing this functionality. Routes from the Cisco SD-WAN overlay would be learned by the MPLS CE router, and as long as the routing policy permits, these routes will be redistributed into the protocol running on the data center core. Conversely, the routes learned on the MPLS CE router from the data center core would be redistributed into BGP and learned by both the WAN Edge router and the service provider. Figure 5-28 details this type of connectivity.

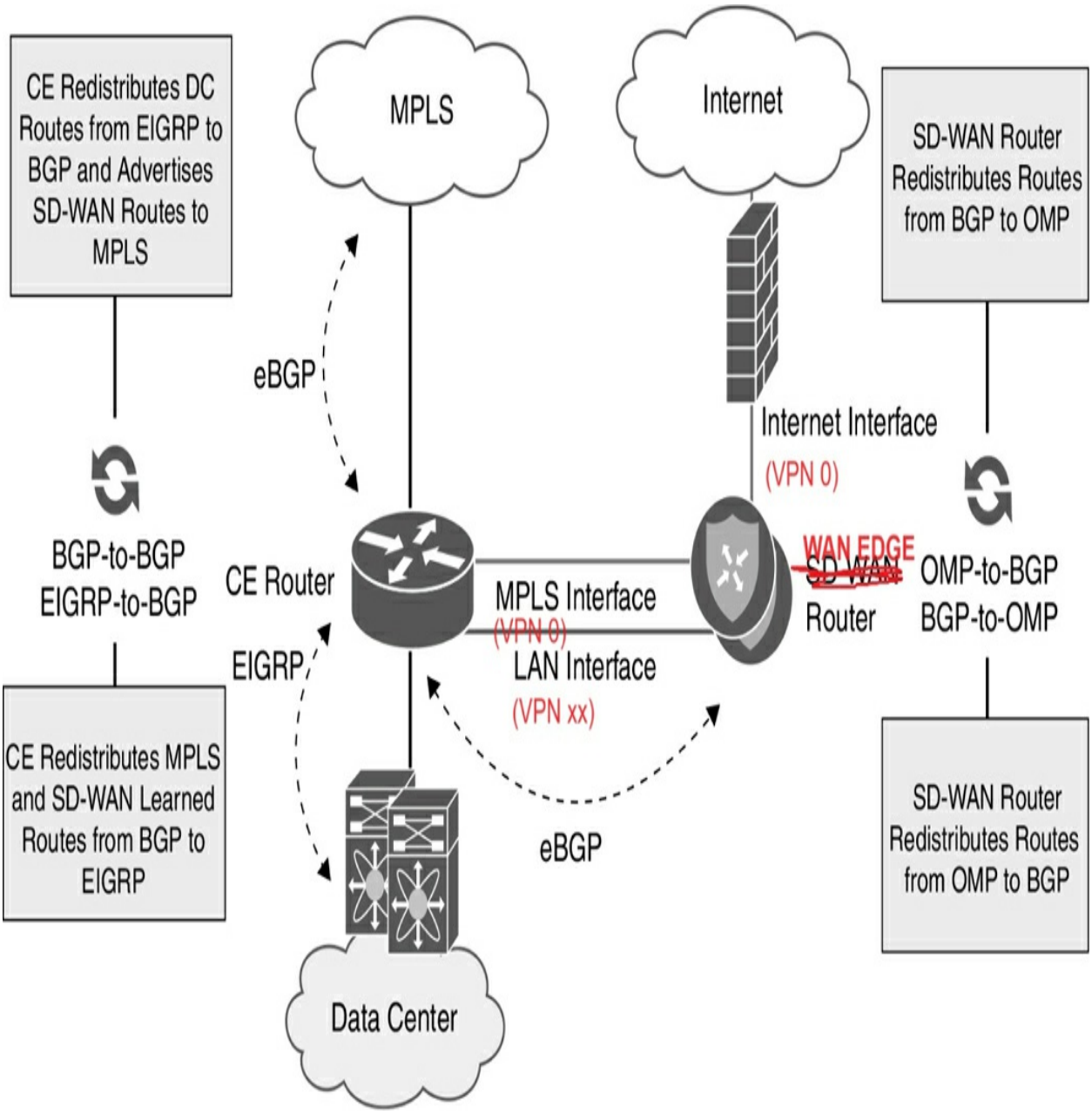


Figure 5-28 *Service-Side CE Integration*

Note

As with any other routing integration exercise, it is important to ensure that path selection occurs as desired, loops are avoided, failover performs as expected, and routing is optimal.

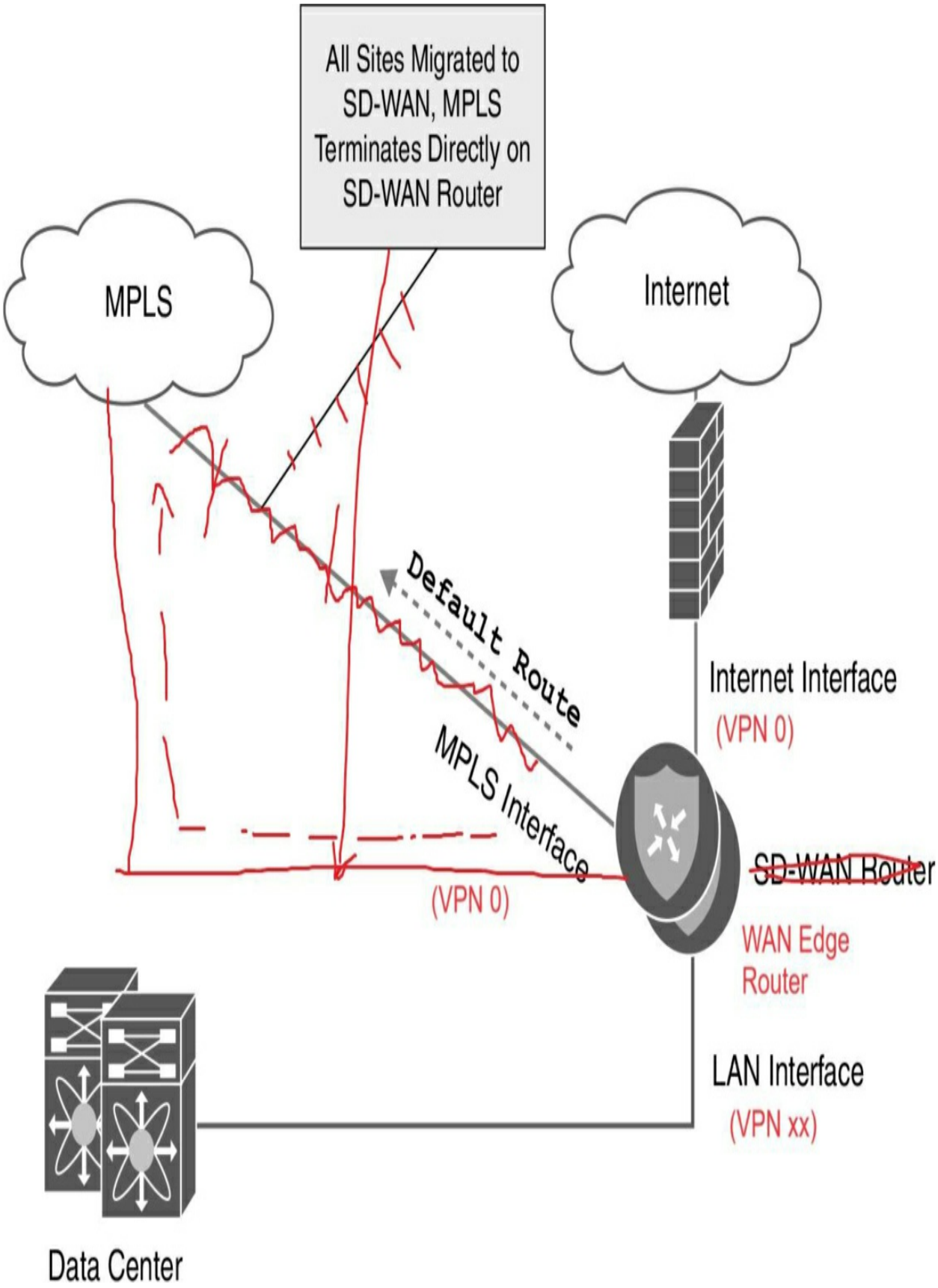
Technet24

To achieve these objectives, leverage the route manipulation, filtering, and tagging tools provided by Cisco SD-WAN.

Keep in mind that while the aforementioned Cisco SD-WAN data center designs are the most recommended and widely deployed, many other methods of providing transport and service-side connectivity do exist and can be investigated for their value, specific to the organization's needs. For instance, subinterfaces can be leveraged for both LAN and WAN connectivity if port density is a concern or if the requirement to transit through a switch or a transparent firewall is applicable. In the end, we are dealing with traditional routing and switching, and many basic design concepts still hold true.

Finally, once all remote sites have been migrated to Cisco SD-WAN, it is possible (but not mandatory) to decommission the legacy MPLS CE routers terminating the transport circuits. The transport circuits need to be re-terminated on the WAN Edge routers directly, which could potentially change the connectivity design and configuration.

In short, connectivity needs to be provided to VPN 0 in some manner. While decommissioning the legacy MPLS CE routers might seem like the right thing to do, many organizations opt to keep them in service to retain compartmentalization and minimize complexity. Figure 5-29 illustrates this potential post-migration design where the MPLS CE router is removed.


All Sites Migrated to SD-WAN, MPLS Terminates Directly on SD-WAN Router
MPLS
Internet
Default Route
Internet Interface
(VPN 0)
MPLS Interface
(VPN 0)
SD-WAN Router
WAN Edge Router
LAN Interface
(VPN xx)
Data Center

Technet24

Figure 5-29 *Direct Transport Termination*

Note

Refer to Cisco's published migration guides and Cisco Validated Design guides for detailed step-by-step instructions on migration and sample configurations.

Cisco SD-WAN Branch Design

Once the data centers have been migrated to Cisco SD-WAN, the branches can follow suit. Cisco SD-WAN branch design can be a bit tricky compared to data center design. Most organizations have branches that come in all shapes and sizes, with different topologies, WAN connectivity models, high availability, and additional services (such as voice and security) that need to be considered. The migration to Cisco SD-WAN could be your opportunity to enforce standardized branch designs and make use of the power of templates or configuration groups. Because it would not be feasible to detail how to migrate every single branch design, this section focuses on the designs that are most common.

Complete CE Replacement: Single WAN Edge Router

The replacement (or upgrade) of a single legacy router to an SD-WAN router requires downtime, as there is no other router to forward traffic during the migration. This design is very straightforward and simply requires you to

terminate the circuits physically or logically on the transport side of the WAN Edge router as well as connect the service side to a LAN core. No routing protocol is required on the transport side, as a default route(s) is used to direct traffic to the respective carrier's next-hop IP address. Both Layer 2 and Layer 3 service-side connectivity is supported. A WAN Edge router can connect to the LAN via 802.1Q subinterfaces and act as the Layer 3 gateway for all of the service-side VLANs. Alternatively, depending on the platform, a WAN Edge router can peer with a Layer 3 LAN core router via BGP, OSPF, EIGRP, or RIPv2. In the Layer 2 design, the connected subnets are automatically redistributed into OMP, and end-to-end reachability is achieved. Figure 5-30 illustrates this type of Layer 2 design.

Technet24

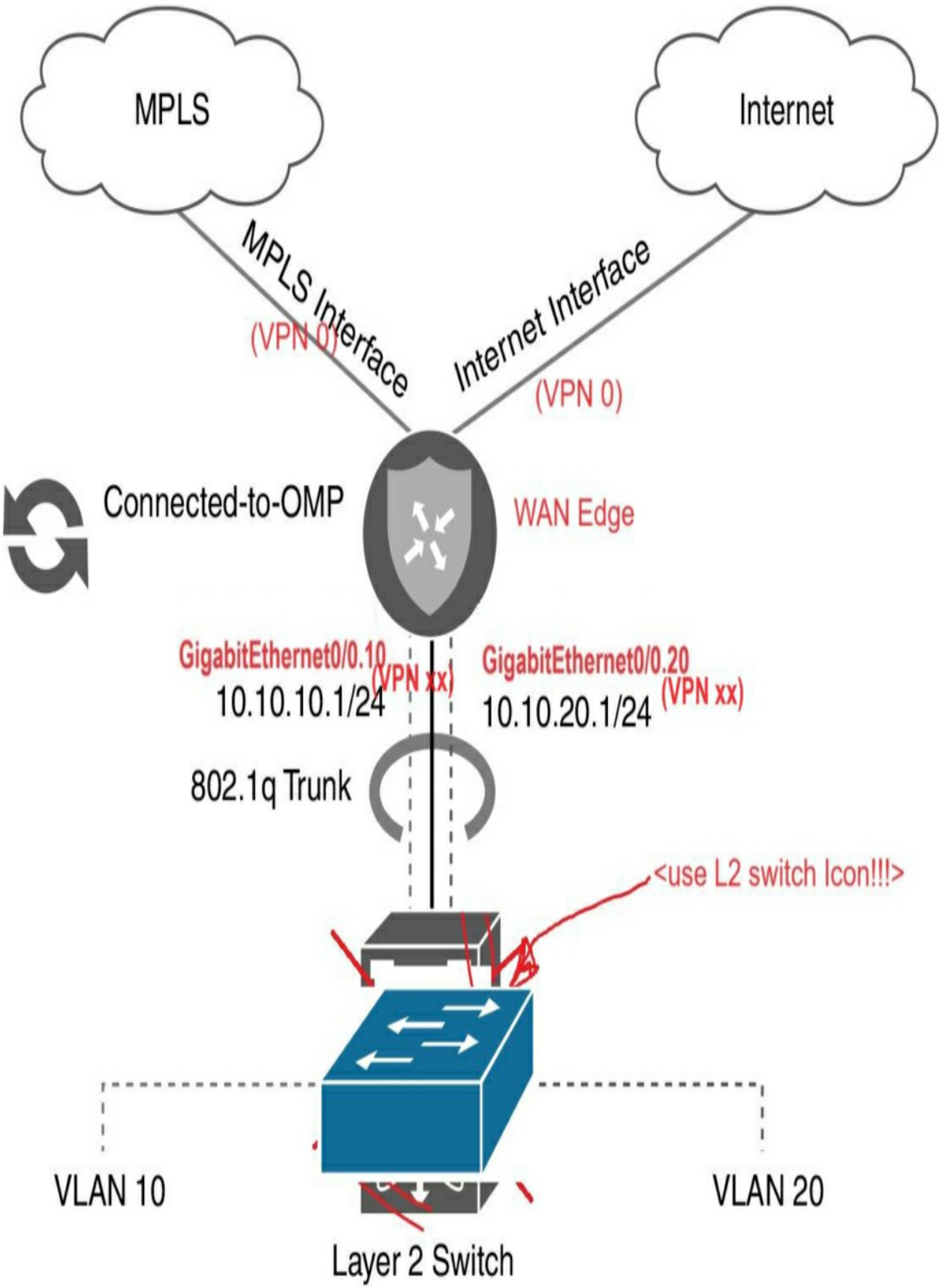

MPLS
Internet
MPLS Interface
(VPN 0)
Internet Interface
(VPN 0)
Connected-to-OMP
WAN Edge
GigabitEthernet0/0.10
(VPN xx)
10.10.10.1/24
GigabitEthernet0/0.20
(VPN xx)
10.10.20.1/24
802.1q Trunk
<use L2 switch Icon!!!>
VLAN 10
VLAN 20
Layer 2 Switch

Figure 5-30 *Complete CE Replacement: Single-Router Layer 2 Branch*

In the Layer 3 design, the routing protocol used on the service side must be redistributed into OMP, and OMP must be redistributed into the service-side routing protocol. Figure 5-31 shows this design.

Technet24

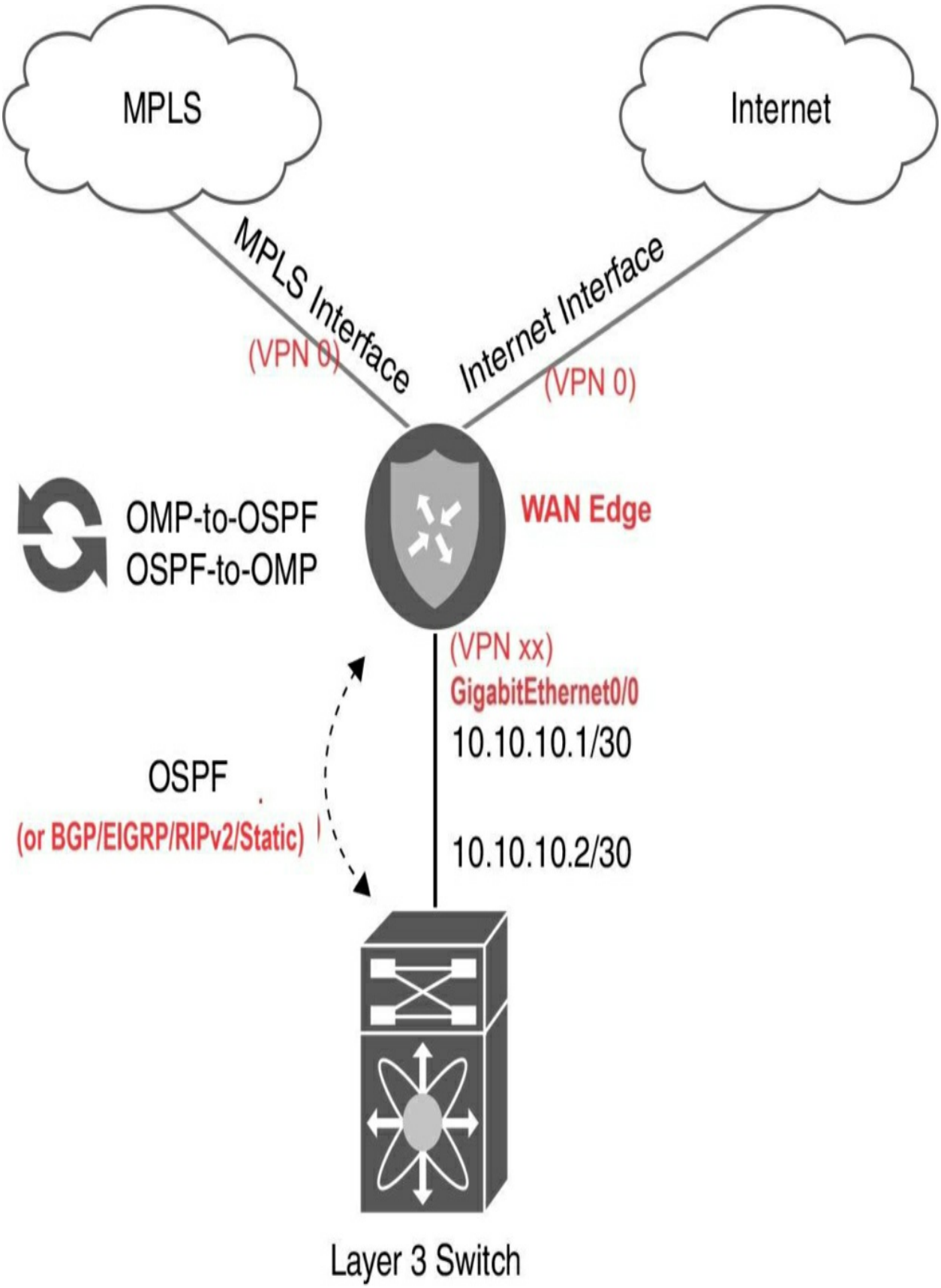

MPLS
Internet
MPLS Interface
(VPN 0)
Internet Interface
(VPN 0)
OMP-to-OSPF
OSPF-to-OMP
WAN Edge
(VPN xx)
GigabitEthernet0/0
10.10.10.1/30
OSPF
(or BGP/EIGRP/RIPv2/Static)
10.10.10.2/30
Layer 3 Switch

Figure 5-31 *Complete CE Replacement: Dual-Router Layer 3 Branch*

Complete CE Replacement: Dual WAN Edge Routers

A branch that requires high availability might have two WAN routers that need to be replaced. In this scenario, it is possible to provide a near hitless cutover since one router can be replaced or upgraded at a time while the other router continues forwarding traffic. Using routing protocol path manipulation and/or VRRP priority adjustment, it would be necessary to first ensure that all the inbound and outbound traffic is shifted to flow over the router that is not initially being upgraded. Once one of the routers is replaced or converted to Cisco SD-WAN, traffic can be shifted over to the newly brought up Cisco SD-WAN overlay network while the other router is then replaced or upgraded.

In this design, illustrated in Figure 5-32, both WAN Edge routers have access to all transports directly—either through multiple handoffs, by utilizing multiple IP addresses and switching infrastructure, or by using *TLOC extensions*.

Technet24

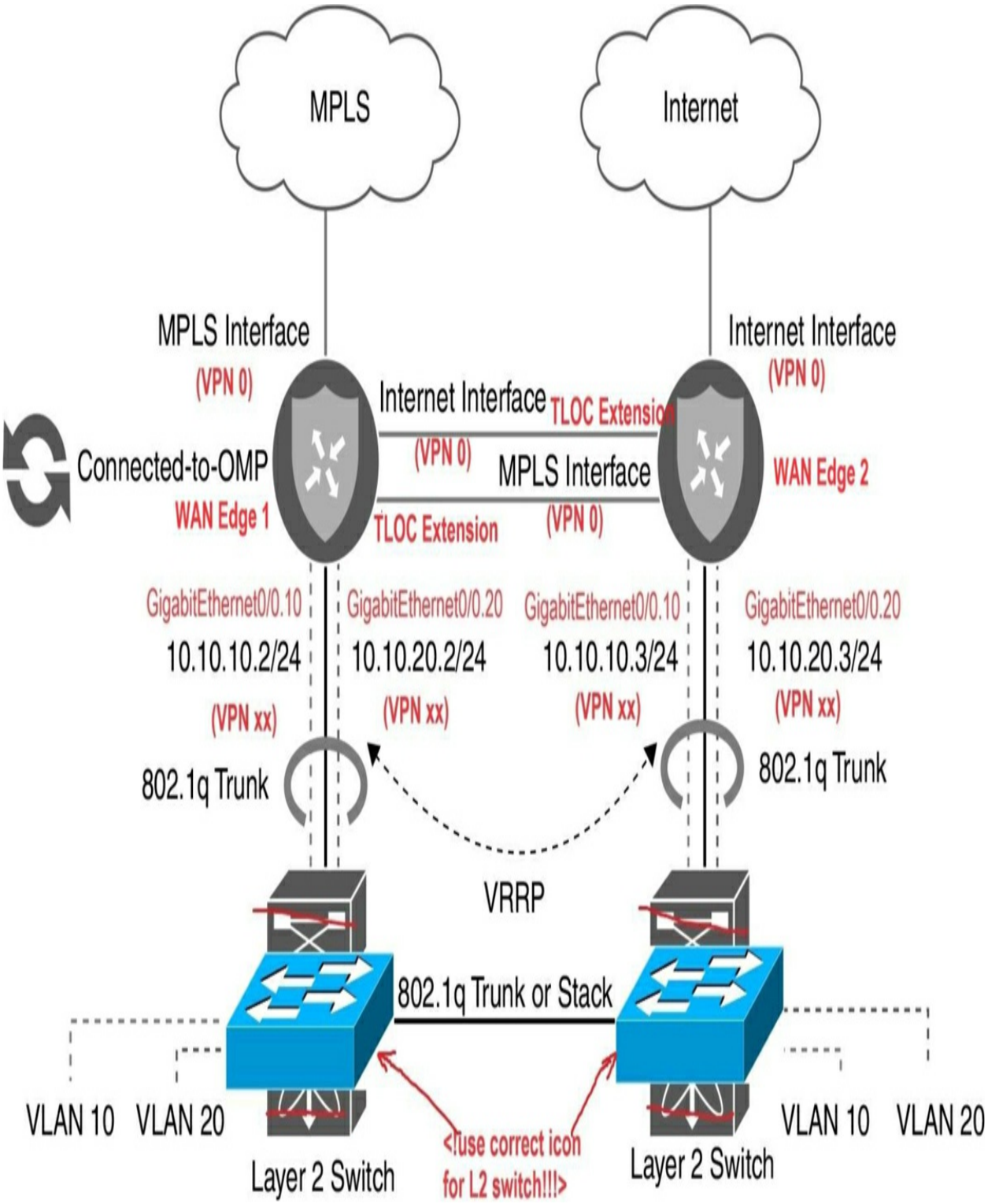
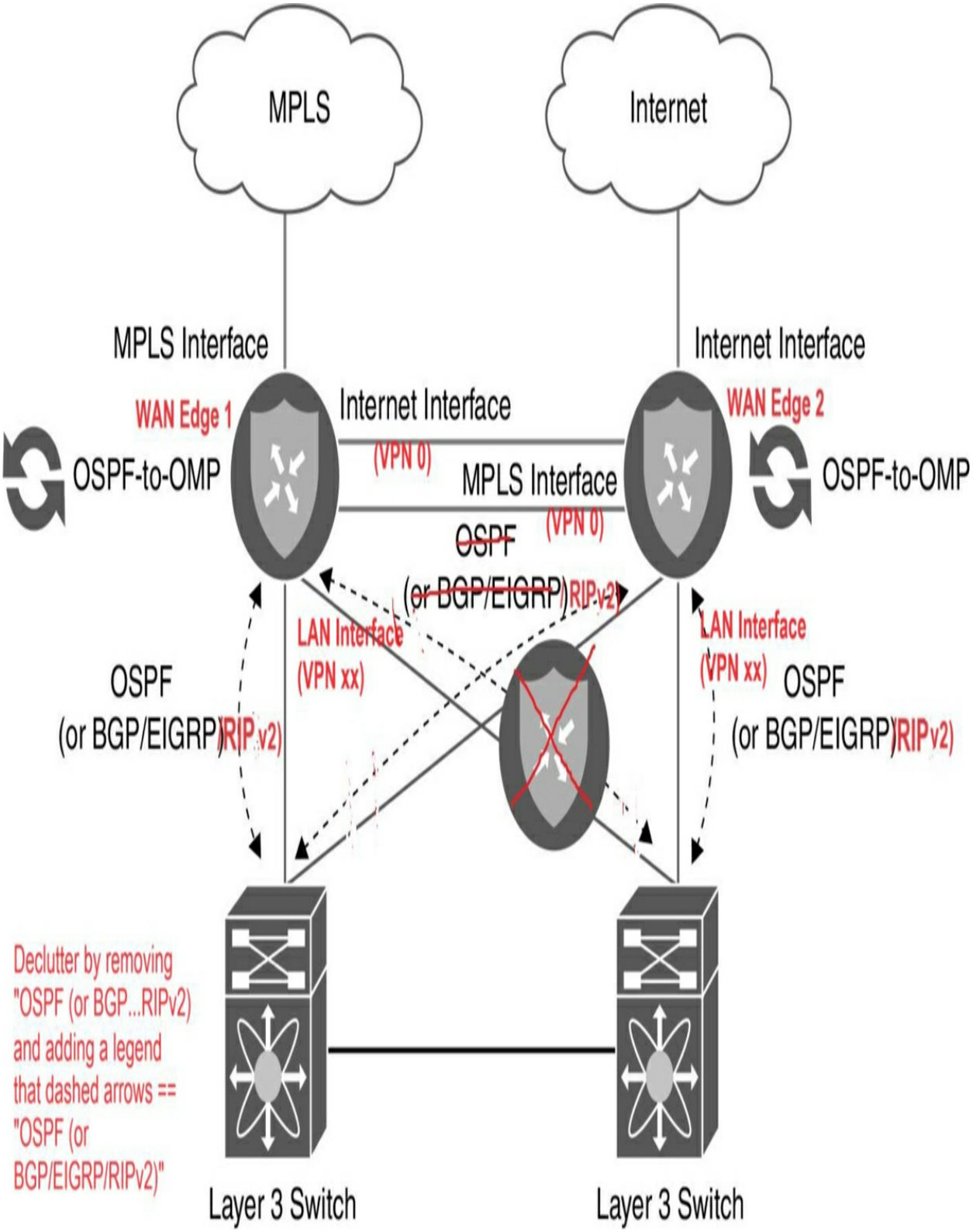


Figure 5-32 *Complete CE Replacement: Dual-Router Layer 2 Branch*

In addition, users have the choice of leveraging a routing protocol instead of

VRRP for service-side integration, as illustrated in Figure 5-33.



Technet24

Figure 5-33 *Complete CE Replacement: Dual-Router Layer 3 Branch*

Integration with an Existing CE Router

Sometimes a branch design calls for the integration of the WAN Edge router with an existing CE router. For example, some MPLS carriers provide and manage the CE router and, as a part of the contract, the router cannot be removed from the branch site. In other cases, the existing CE router may be providing other services for the branch that must stay intact post-migration (such as voice gateway functionality). In order for the WAN Edge router to have connectivity into the MPLS underlay, the design must be made such that the WAN Edge router can tunnel through the existing MPLS CE router. This can easily be accomplished by connecting a physical or logical transport-side interface, configured as a private TLOC in a /30 transit network, to an available interface or subinterface on the existing CE router. The WAN Edge router would then be configured with a default route pointing to the next-hop IP address of the MPLS CE router. In addition, to provide connectivity to the tunnel endpoint IP address, the MPLS CE router needs to advertise this transit network into the underlay network via a routing protocol such as BGP. Figure 5-34 illustrates this scenario.

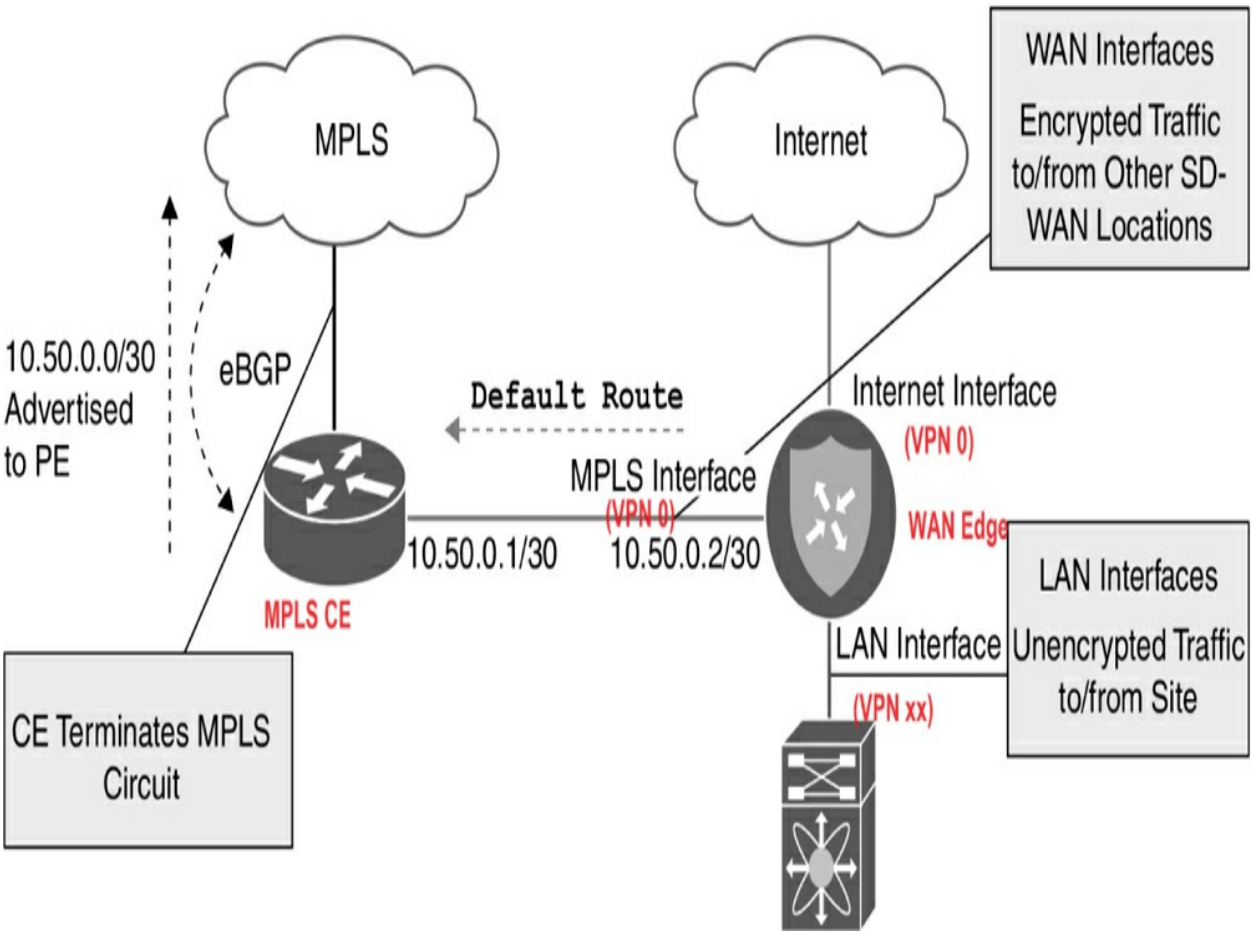


Figure 5-34 *Branch CE Integration*

Integration with a Branch Firewall

It isn’t uncommon for a branch site to have a firewall terminating Internet connectivity and providing services such as direct Internet access, secure access control, zone-based segmentation, URL filtering, NAT, IDS/IPS, and so on. Because of the routing DNA embedded in Cisco SD-WAN, there are all sorts of methods for integrating with a branch firewall, depending on the desired traffic flow and traffic visibility requirements.

Firewalls can be inserted in front of the WAN Edge router, in parallel with

Technet24

the WAN Edge router, or even behind the WAN Edge router in either Layer 3 routed or Layer 2 transparent mode. The appropriate integration model depends on several factors, such as physical connectivity, logical traffic flow, traffic visibility requirements, segmentation architecture, and NAT requirements. The only requirement for the WAN Edge router is that it is provided transport-side Internet connectivity for control and data plane tunnels as well as service-side LAN connectivity to receive and forward user traffic into and out of the overlay.

If all Internet-bound traffic is backhauled to a data center or hub egress point, firewall integration is very straightforward. In this design, the firewall terminates the Internet connection directly and provides access to the WAN Edge router through either an allocated public IP address in a DMZ or through NAT (one-to-one NAT or PAT).

Note

While Cisco SD-WAN supports tunneling through several types of NAT, some dynamic NAT and dynamic PAT combinations may cause tunnel establishment failures (especially if that type of NAT is on both sides of the tunnel). Refer to the “Network Address Translation” section in Chapter 3, “Control Plane and Data Plane Operations,” for more details on supported NAT configurations.

Secure direct Internet access is one of the key benefits of migrating to Cisco SD-WAN, and the design options discussed in this section provide this functionality.

In some networks, it is beneficial for a firewall to provide direct Internet access functionality to clients since it may be running advanced security services such as URL filtering. With this design, Internet-bound traffic ingresses the service side of the WAN Edge router and, instead of performing NAT and local egress itself, the WAN Edge router forwards the traffic over another service-side link to the adjacent firewall. The firewall can then perform inspection, access control, and NAT prior to forwarding the traffic to the Internet. Traffic to private network destinations follows the Cisco SD-WAN overlay. This is most optimal when the Internet handoff has more than one physical port. Figure 5-35 details this design.

Technet24

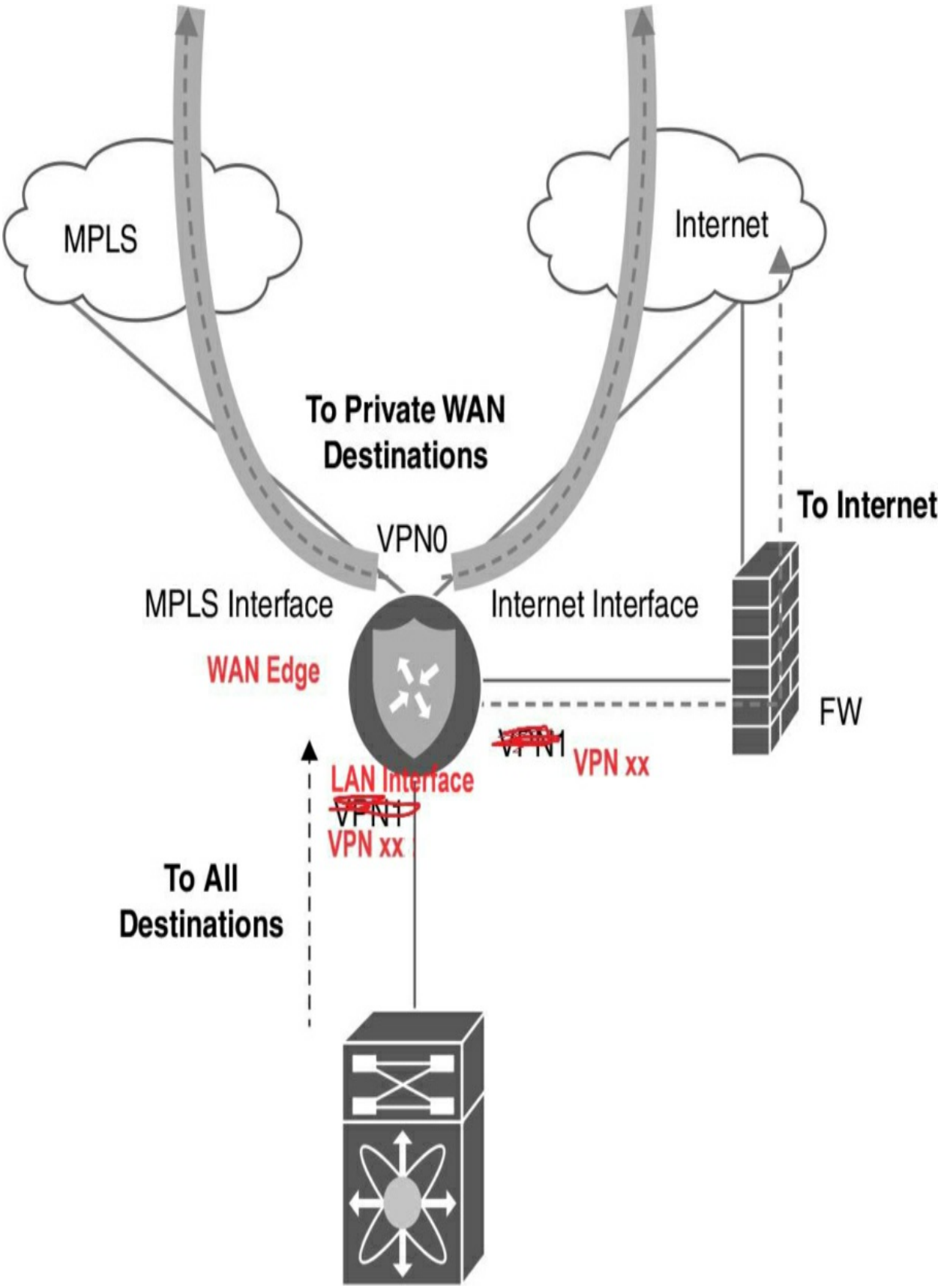

MPLS
Internet
To Private WAN Destinations
To Internet
VPN0
MPLS Interface
Internet Interface
WAN Edge
FW
VPN xx
LAN Interface
VPN xx
To All Destinations

Figure 5-35 *Direct Internet Access Through a Firewall*

If a dedicated handoff to the Internet is not available, an alternative design would be to connect the transport side of the WAN Edge router to the firewall. In this way, the WAN Edge router has access to the Internet for control and data plane tunnels while also keeping a separate service-side connection to forward Internet-bound traffic to the firewall for processing and local egress. Figure 5-36 illustrates this design.

Technet24

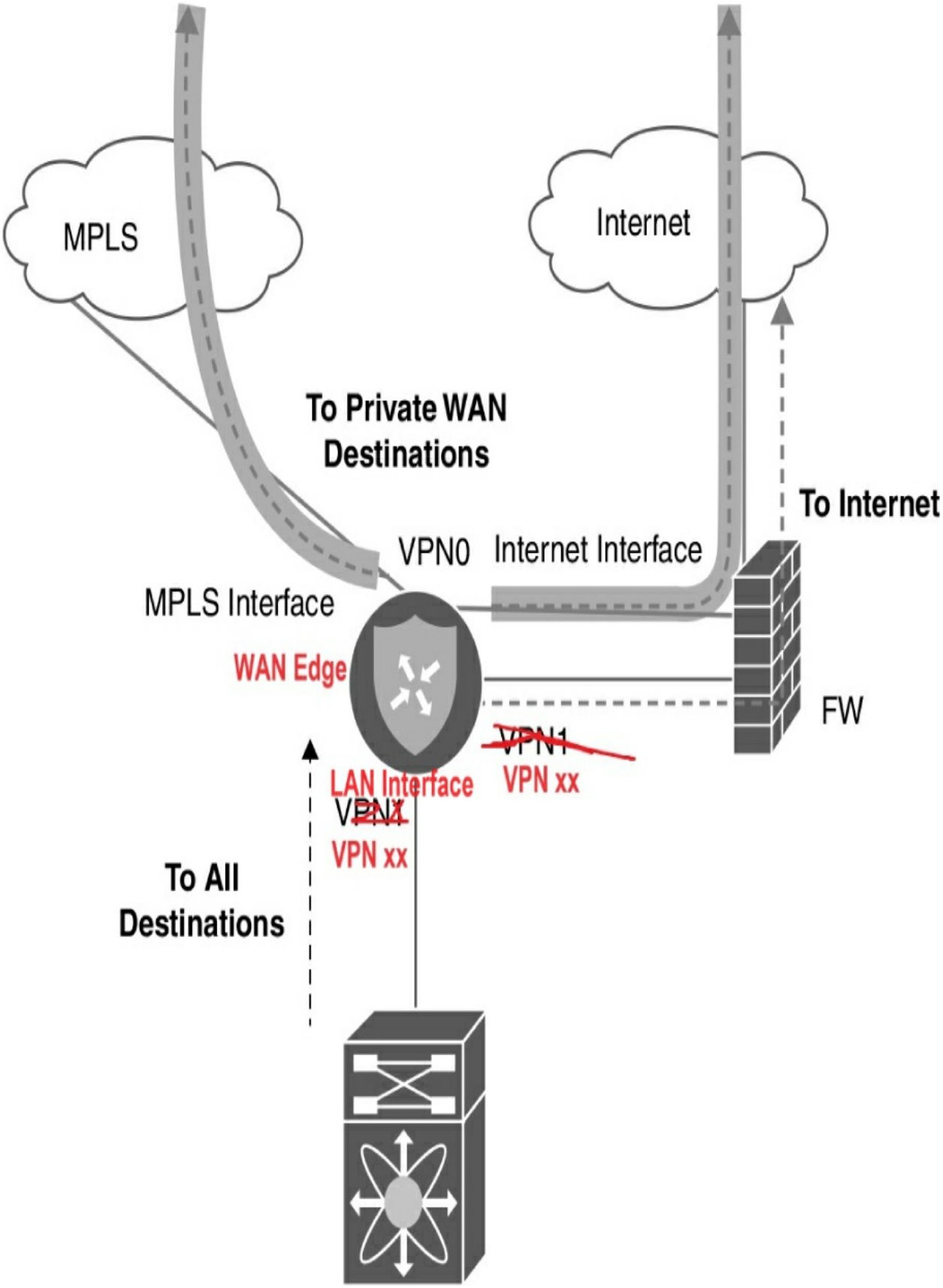

MPLS
Internet
To Private WAN Destinations
To Internet
VPN0
Internet Interface
MPLS Interface
WAN Edge
FW
VPN1
VPN xx
LAN Interface
VPN1
VPN xx
To All Destinations

Figure 5-36 *Transport and Direct Internet Access Through a Firewall*

Finally, if you want all traffic to flow through the WAN Edge router and only a single connection exists between the router and the firewall, consider using a single transport-side connection is for both tunneling and direct Internet access. NAT is configured on the transport-side interface connected to the firewall, and Cisco SD-WAN policy can selectively break out Internet-bound traffic to the underlay. The firewall receives this Internet-bound traffic and may perform NAT a second time, as it sends the traffic to the Internet. It is important to note that because the WAN Edge router is performing NAT (which is a requirement for WAN Edge routers to break out traffic locally), the firewall loses visibility into the original client source IP addresses. Figure 5-37 shows this design.

Technet24

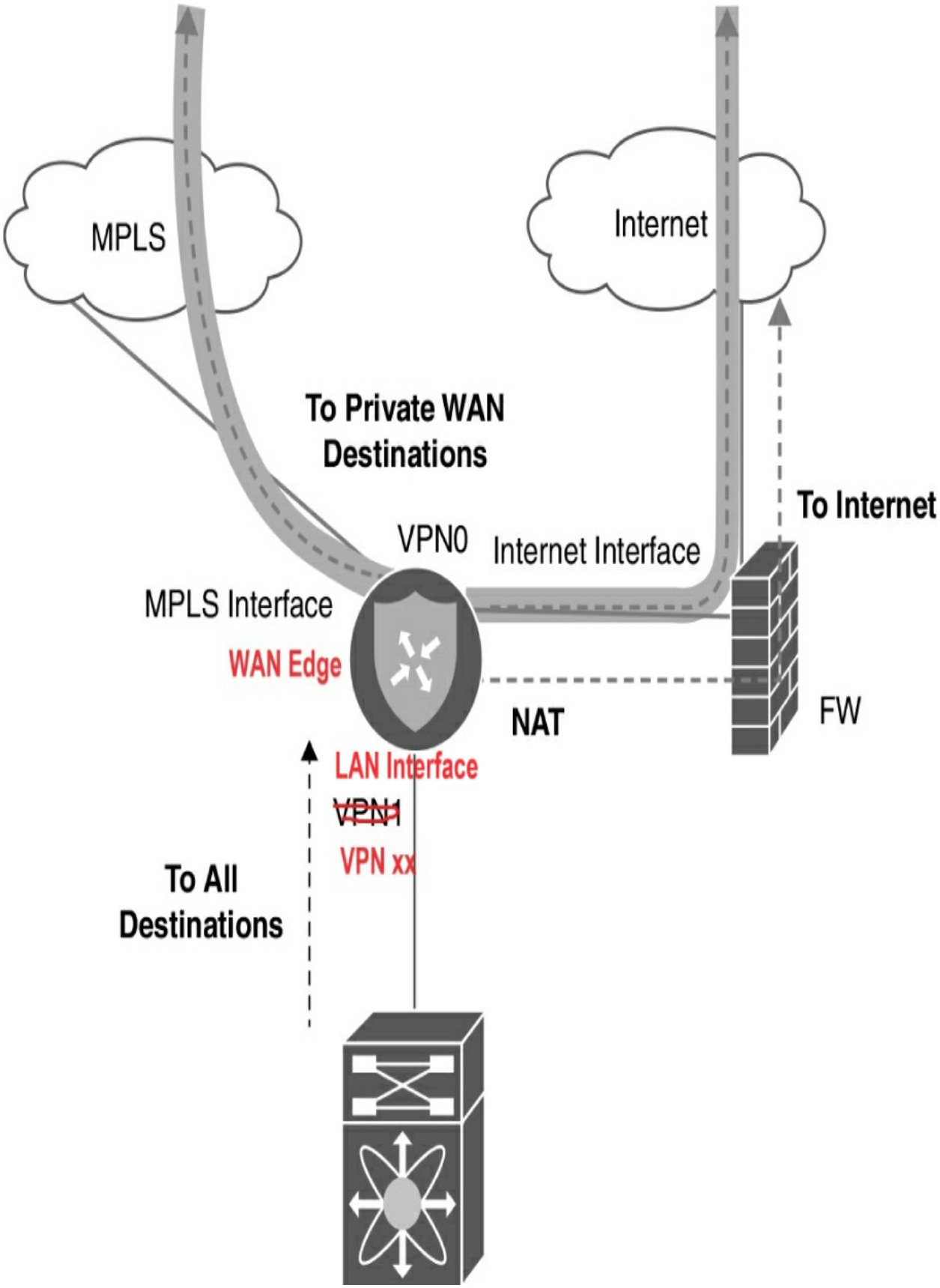

MPLS
Internet
To Private WAN Destinations
To Internet
VPN0
Internet Interface
MPLS Interface
WAN Edge
NAT
FW
LAN Interface
VPN1
VPN xx
To All Destinations

Figure 5-37 *Transport and Direct Internet Access Through a Firewall's Single Link*

Integrating Cisco SD-WAN with Existing Networks



This section details how integration between the existing network and SD-WAN overlay networks can be achieved through several design options.

Overlay Only

To minimize complexity and the potential for routing loops, it is recommended to adhere to the designs presented earlier in this chapter for data center and branch migration. These designs have been proven time and time again to provide a graceful migration solution for organizations of all sizes, complexities, and application types. Overlay-only migrations provide a clear and deterministic traffic flow and are easy to control, scale, and troubleshoot. Figure 5-38 shows traffic flow between a Cisco SD-WAN site and a legacy non-SD-WAN site with this design in place.

Technet24

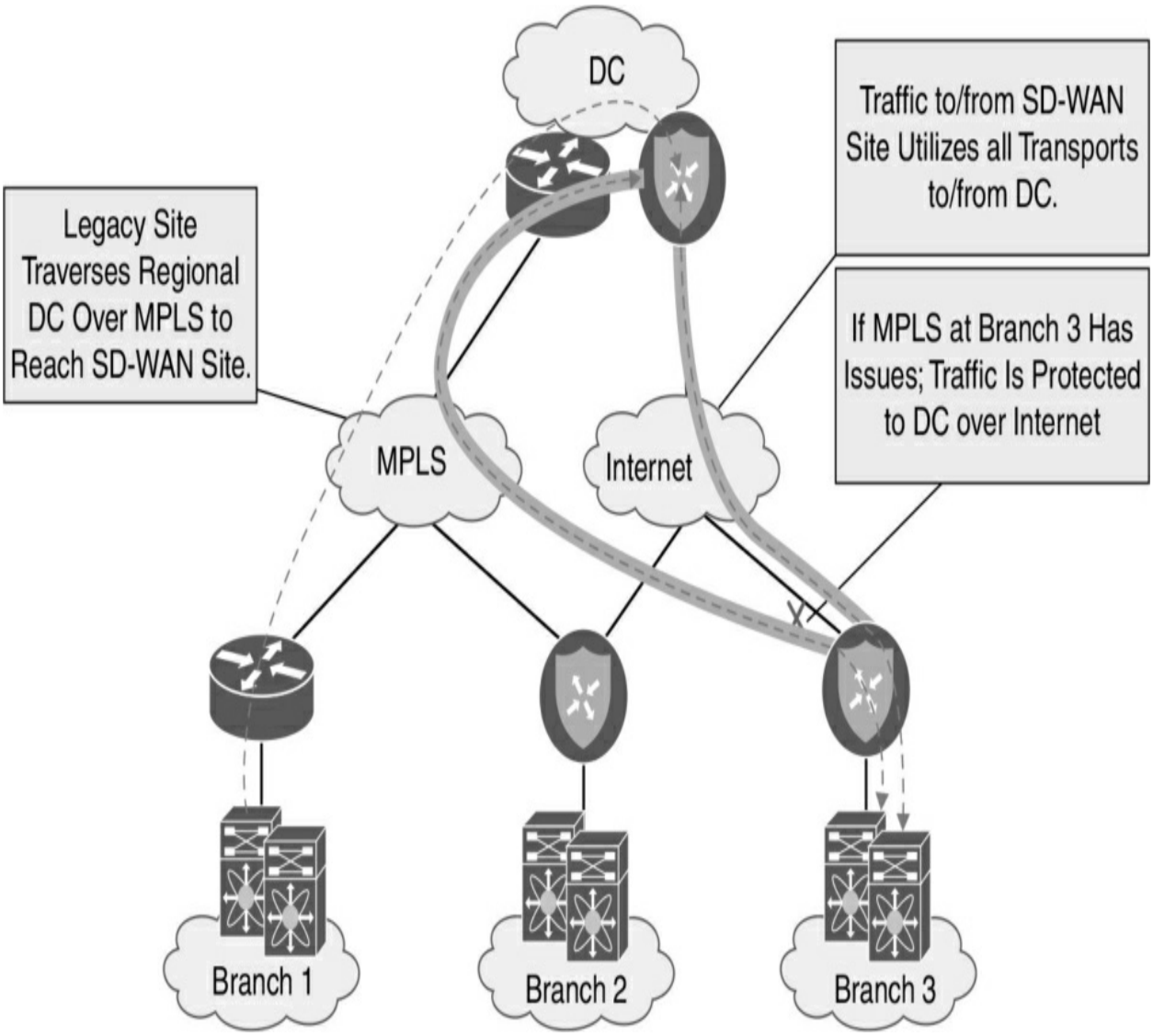


Figure 5-38 *SD-WAN to Non-SD-WAN Site Overlay-Only Traffic Flow*

Understanding the traffic flow from an SD-WAN site to another SD-WAN site is also critical. Figure 5-39 highlights the traffic flow patterns experienced when using an overlay-only design.

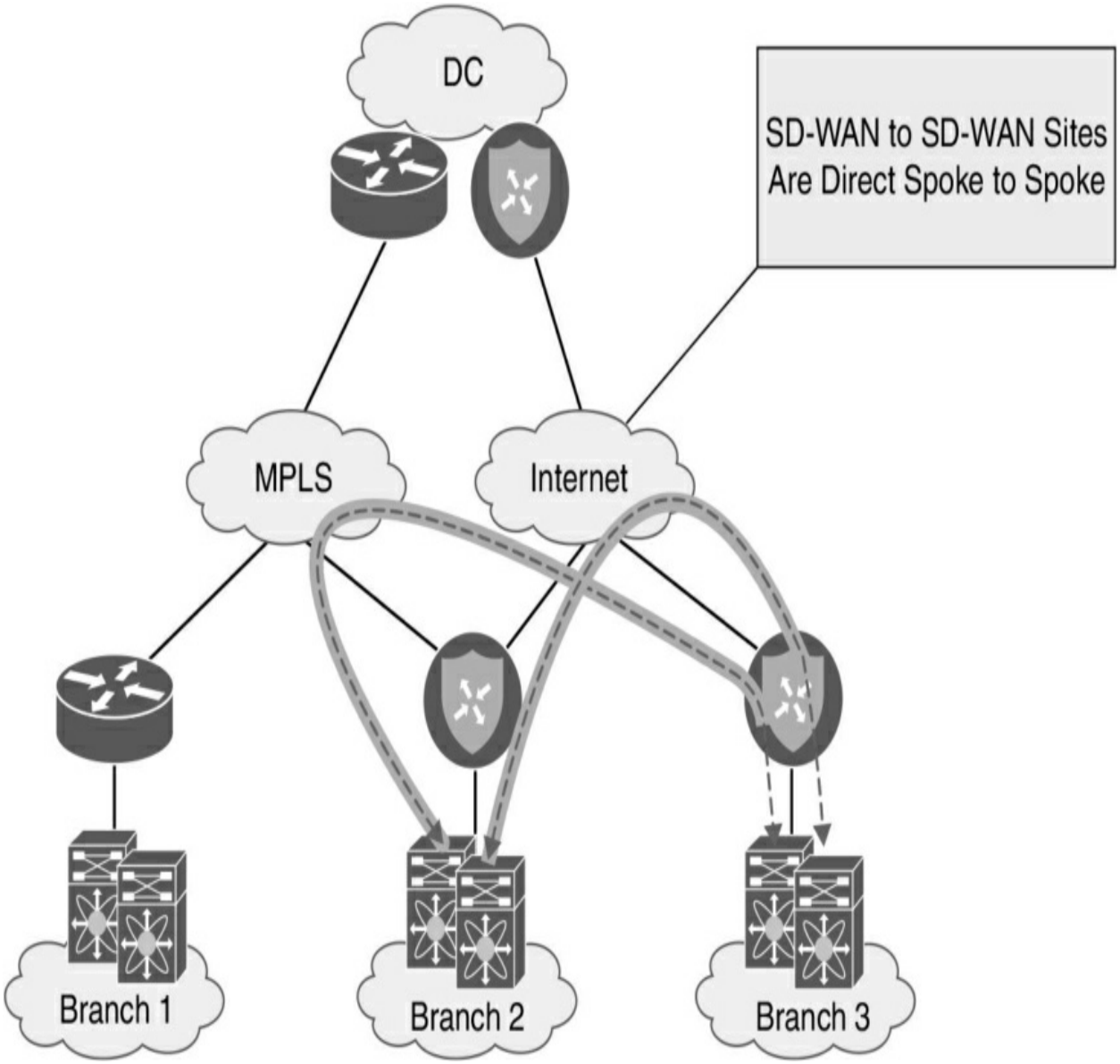


Figure 5-39 *SD-WAN to SD-WAN Site Overlay-Only Traffic Flow*

Overlay with Underlay Backup

If a branch site leveraging MPLS and Internet for the Cisco SD-WAN overlay has only a single WAN Edge router used alongside the MPLS CE router, another design opportunity exists that can allow for a backup path to become active if the WAN Edge router fails. For this design to work

Technet24

properly, routing needs to be manipulated in such a way that the site prefixes advertised by the MPLS CE router into the underlay are less preferred in the network than those being advertised by the WAN Edge router into the overlay. Conversely, from the perspective of the LAN, the WAN Edge router should be the preferred path to reach remote networks. Figure 5-40 illustrates this.

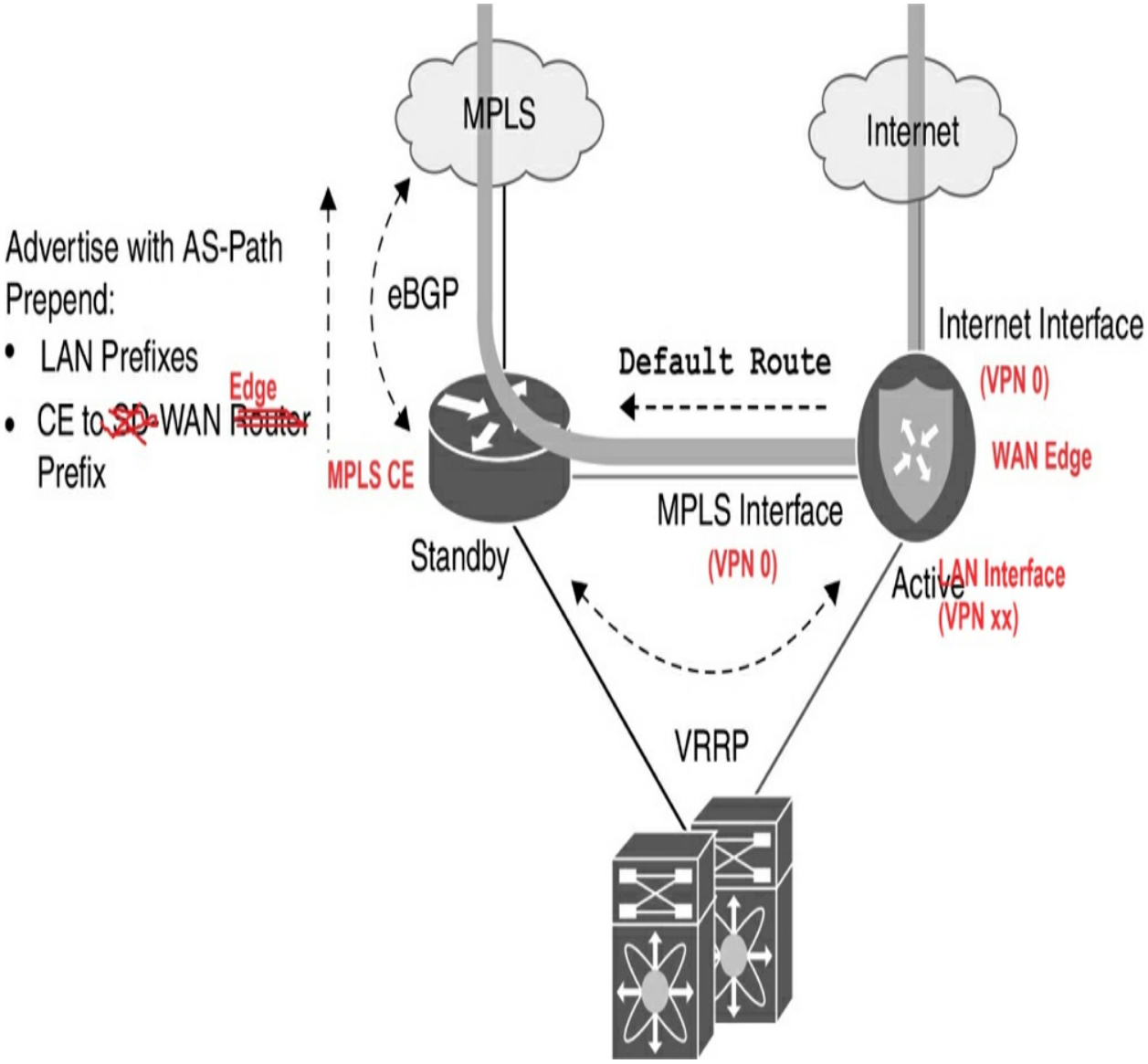


Figure 5-40 *Overlay with Underlay Backup: Layer 2 Design*

This manipulation of path preference ensures that the traffic flow stays symmetric over the Cisco SD-WAN fabric and only utilizes the underlay if the WAN Edge router were to fail. One way to manipulate path preference for the underlay is to only advertise local site prefixes from the MPLS CE router to the service provider and use BGP AS path prepending. Figure 5-41 shows this type of scenario. Naturally, the data center will advertise the remote site prefixes learned from the data center WAN Edge router into the underlay with a shorter AS path, ensuring a more preferred, symmetric flow.

Technet24

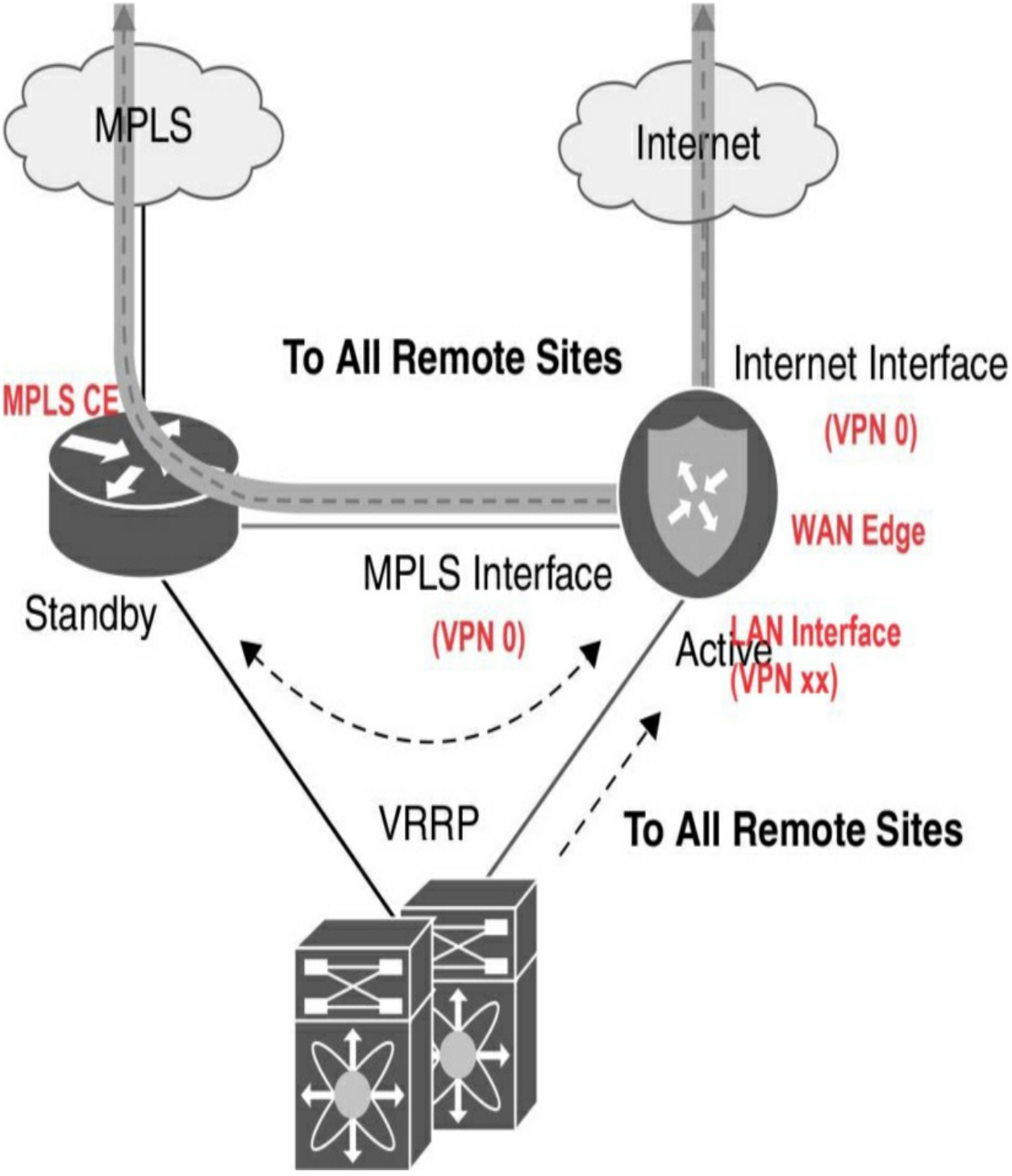


Figure 5-41 *Overlay with Underlay Backup: Layer 2 Design Traffic Flow*

It should be noted that LAN-side routing also needs to be manipulated in order to prefer the WAN Edge router (for normal operation) and fail over to

the MPLS CE router during an impaired state. This can be accomplished in a Layer 2 branch by running VRRP between the WAN Edge router and the MPLS CE router (with the WAN Edge router having the higher priority). Figure 5-42 shows this design option.

Technet24

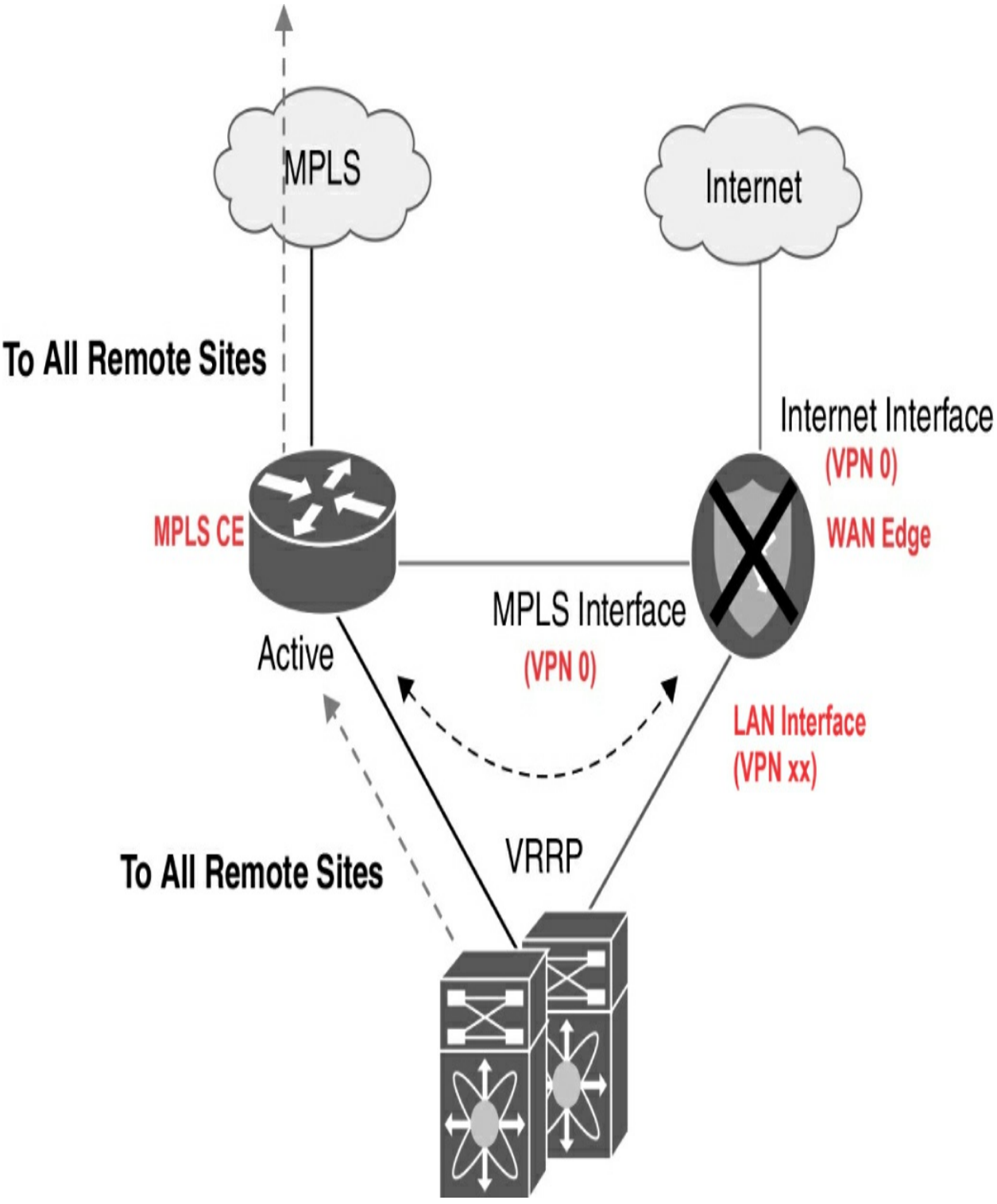


Figure 5-42 *Overlay with Underlay Backup: Layer 2 Design Traffic Flow During Failover*

In a Layer 3 branch design, advertising remote prefixes to the LAN with a more attractive metric from the WAN Edge router is also an option, as illustrated in Figure 5-43.

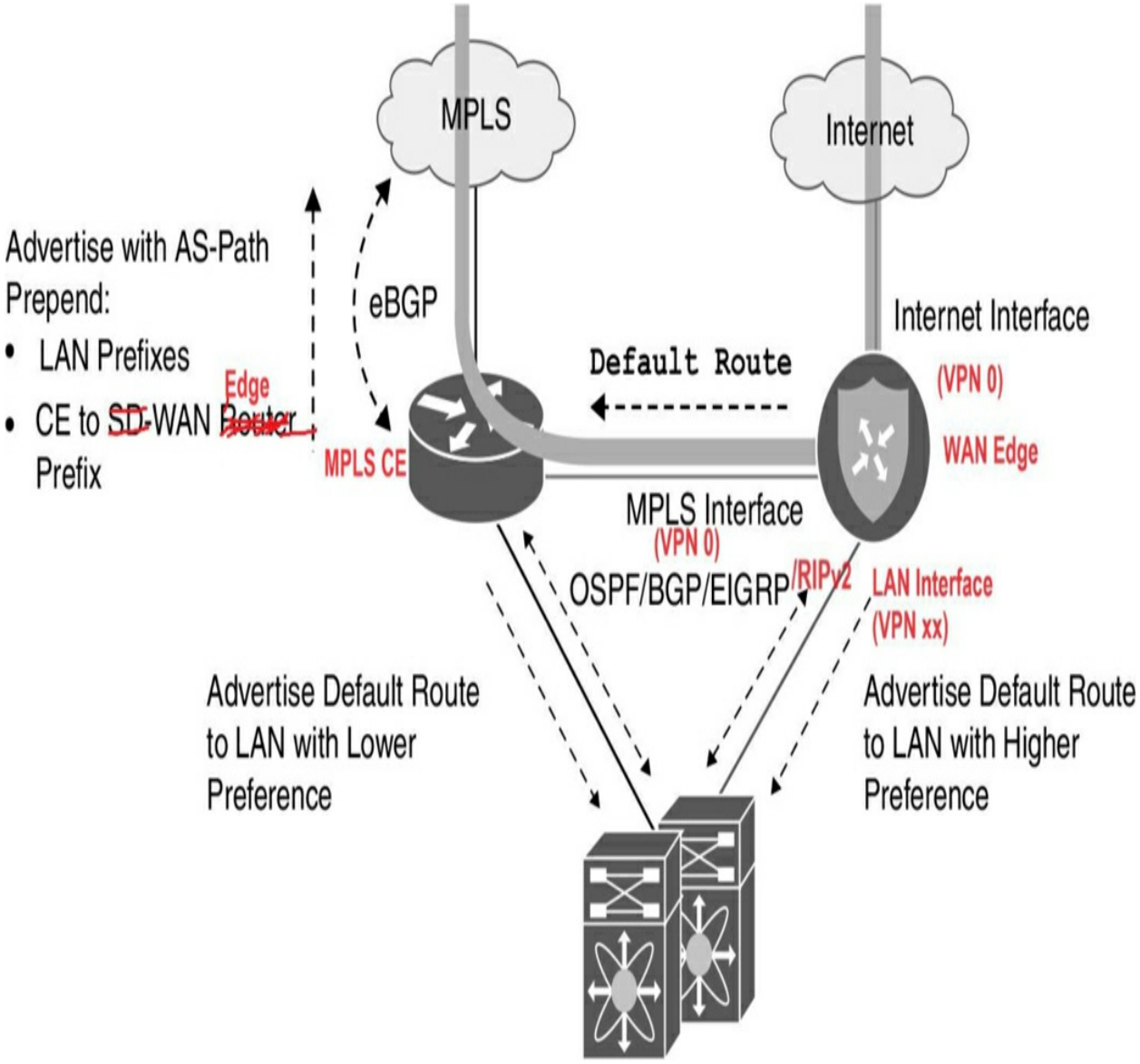


Figure 5-43 *Overlay with Underlay Backup: Layer 3 Design*

Finally, for the underlay backup architecture to work correctly, it is important to do some filtering at the data center. To ensure that remote branch routes

Technet24

are learned and preferred through the overlay (and asymmetry or route looping is avoided), create an outbound filter toward the WAN Edge router to limit the learned routes to those originating from the data center. Make sure to also advertise a default route or summary routes into the overlay. Figure 5-44 shows this design option.

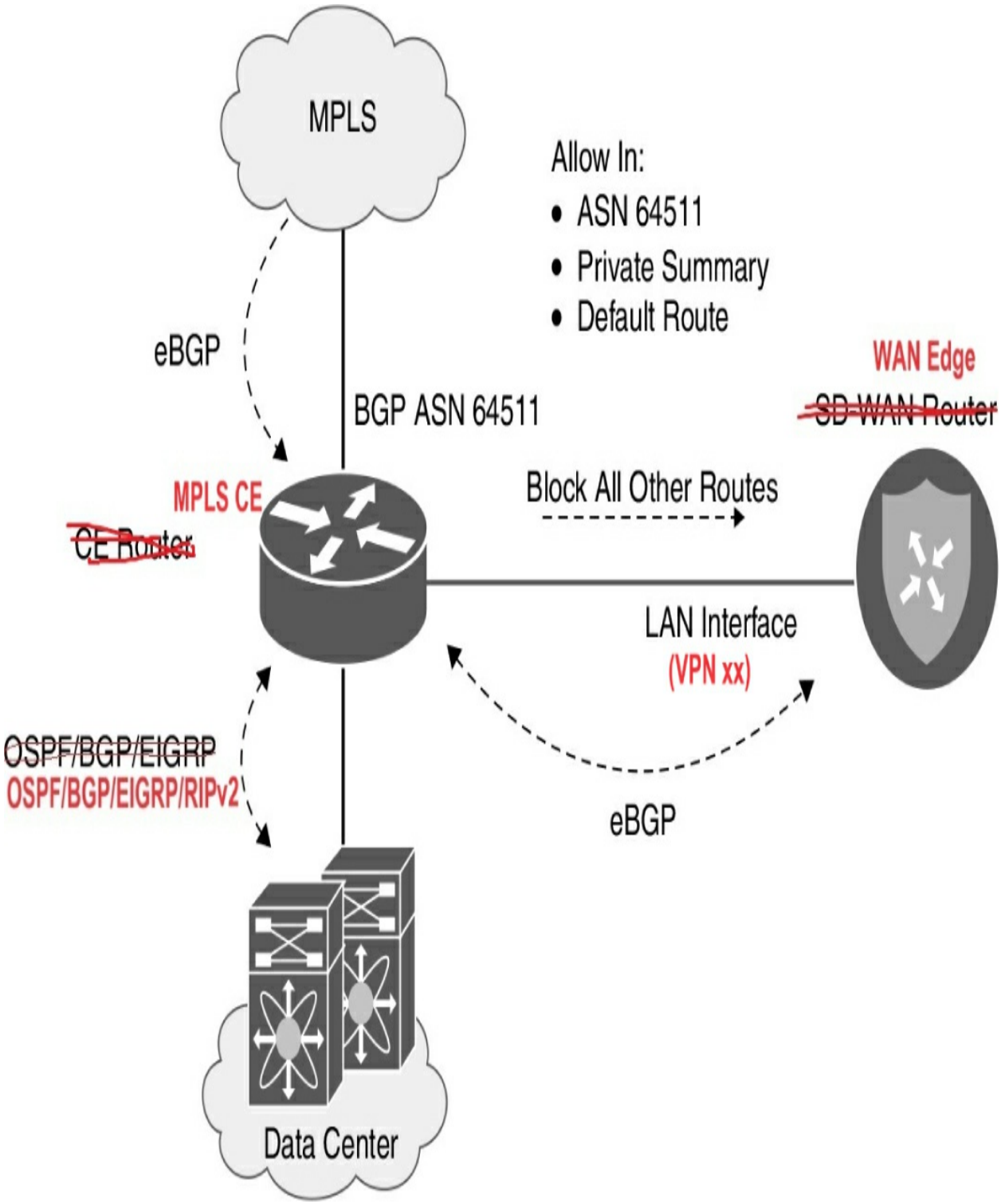


Figure 5-44 *Overlay with Underlay Backup: Data Center Considerations*

Figure 5-45 reviews the traffic flow patterns experienced when using an

Technet24

overlay with an underlay backup design.

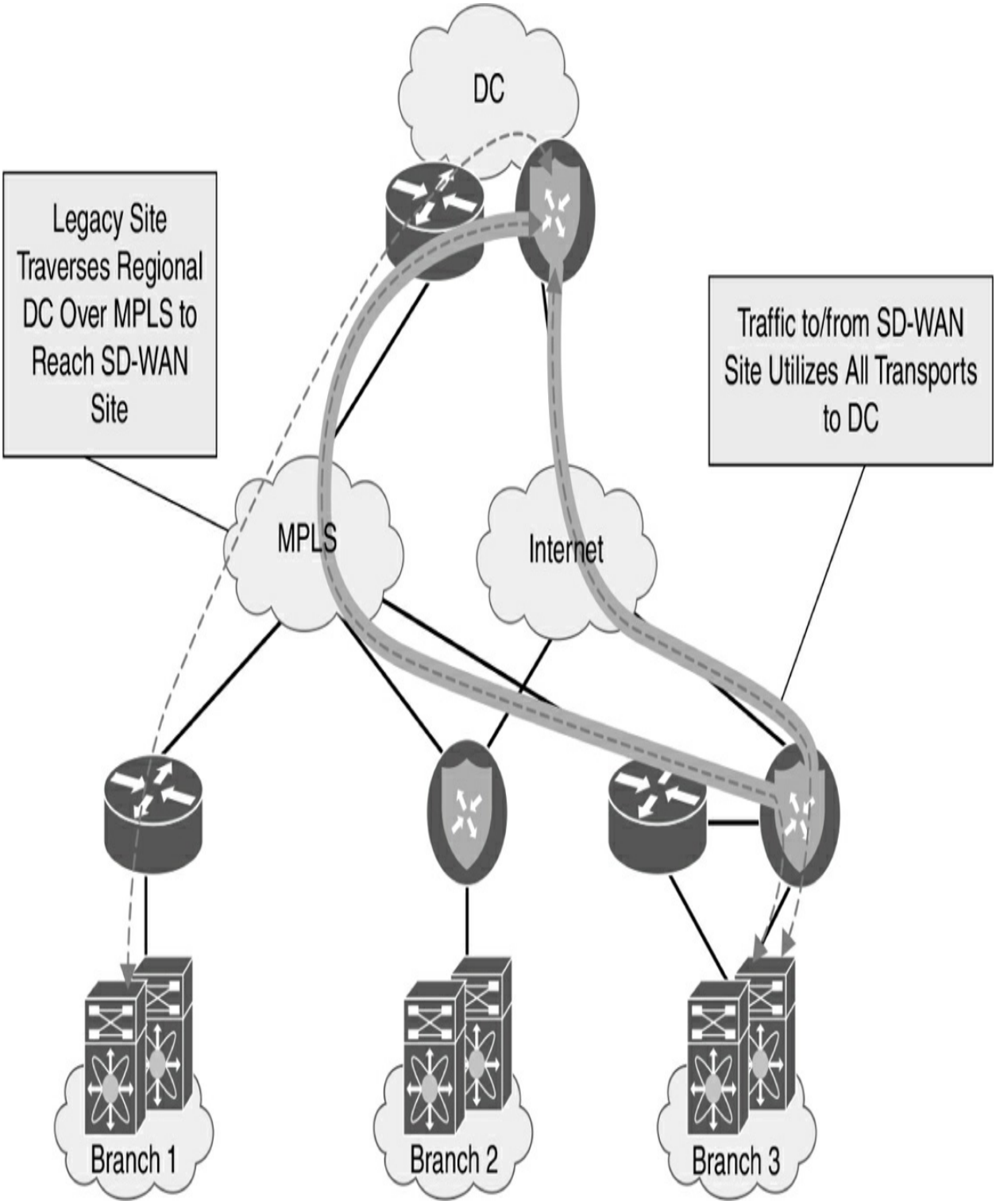
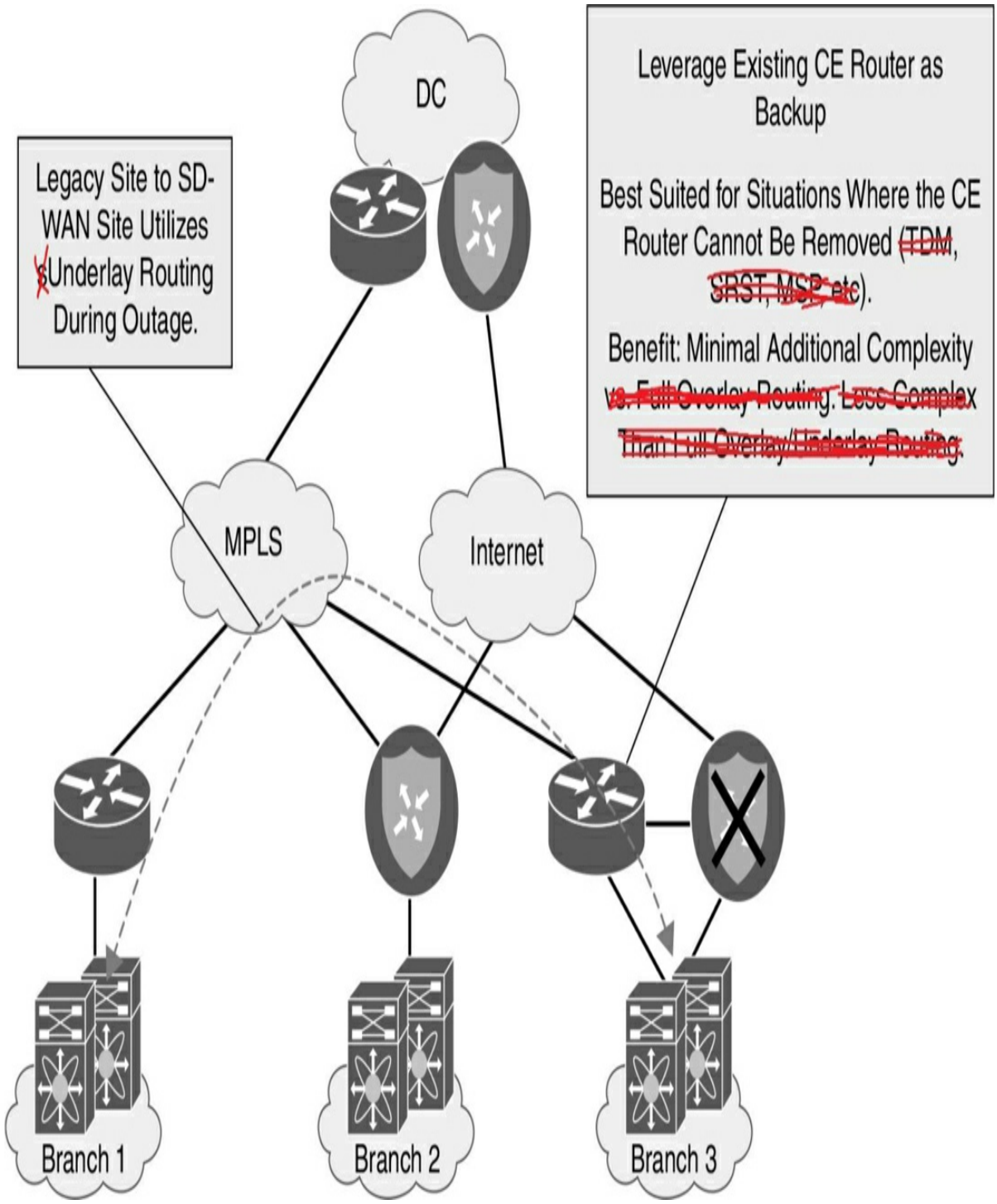


Figure 5-45 *Overlay with Underlay Backup Traffic Flow*

It is also important to see the traffic flow during a failover condition. This helps visualize the path the traffic will take during a failure scenario. Figure 5-46 highlights the backup traffic flow during a failover event.



Technet24

Figure 5-46 *Overlay with Underlay Backup Traffic Flow During Failover*

Full Overlay and Underlay Integration

For some organizations, due to strict application latency requirements, certain branches migrated to Cisco SD-WAN may need to communicate directly with non-migrated branches through a lower-latency underlay path. For example, CIFS traffic, which is relatively sensitive to latency, may suffer from going through additional hops at the data center for transit to another site. To address this latency requirement, SD-WAN network can be architected to use the overlay path for communication with migrated sites and the underlay path for communication with non-migrated sites, enabling full overlay and underlay integration. Figure 5-47 shows routing from a non-SD-WAN site to a Cisco SD-WAN site in this scenario. Figure 5-48 illustrates the traffic flow between two Cisco SD-WAN sites while leveraging a full overlay and underlay integration design.

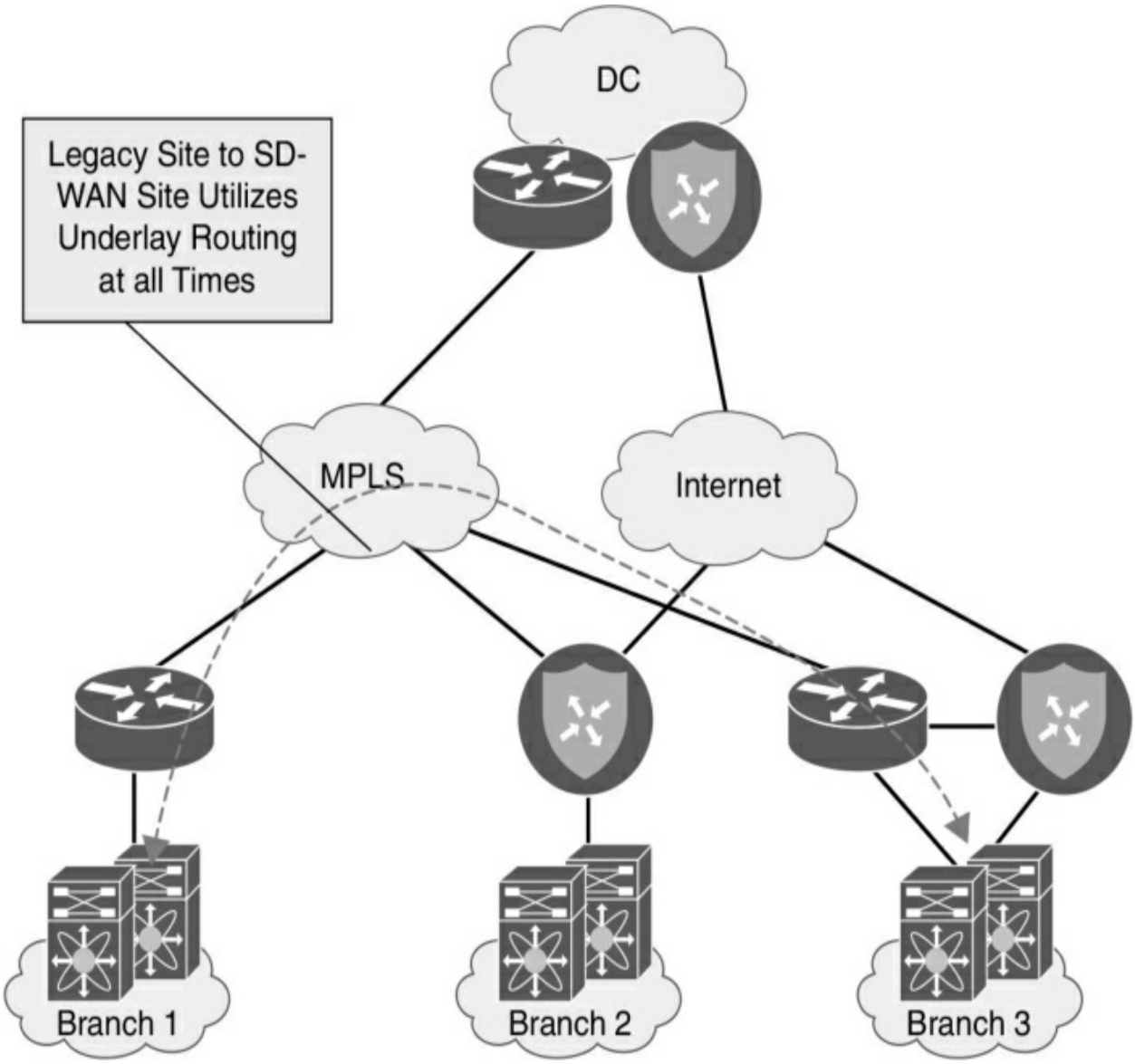


Figure 5-47 *Full Overlay with Underlay Routing: Legacy Site to SD-WAN Site Traffic Flow*

Technet24

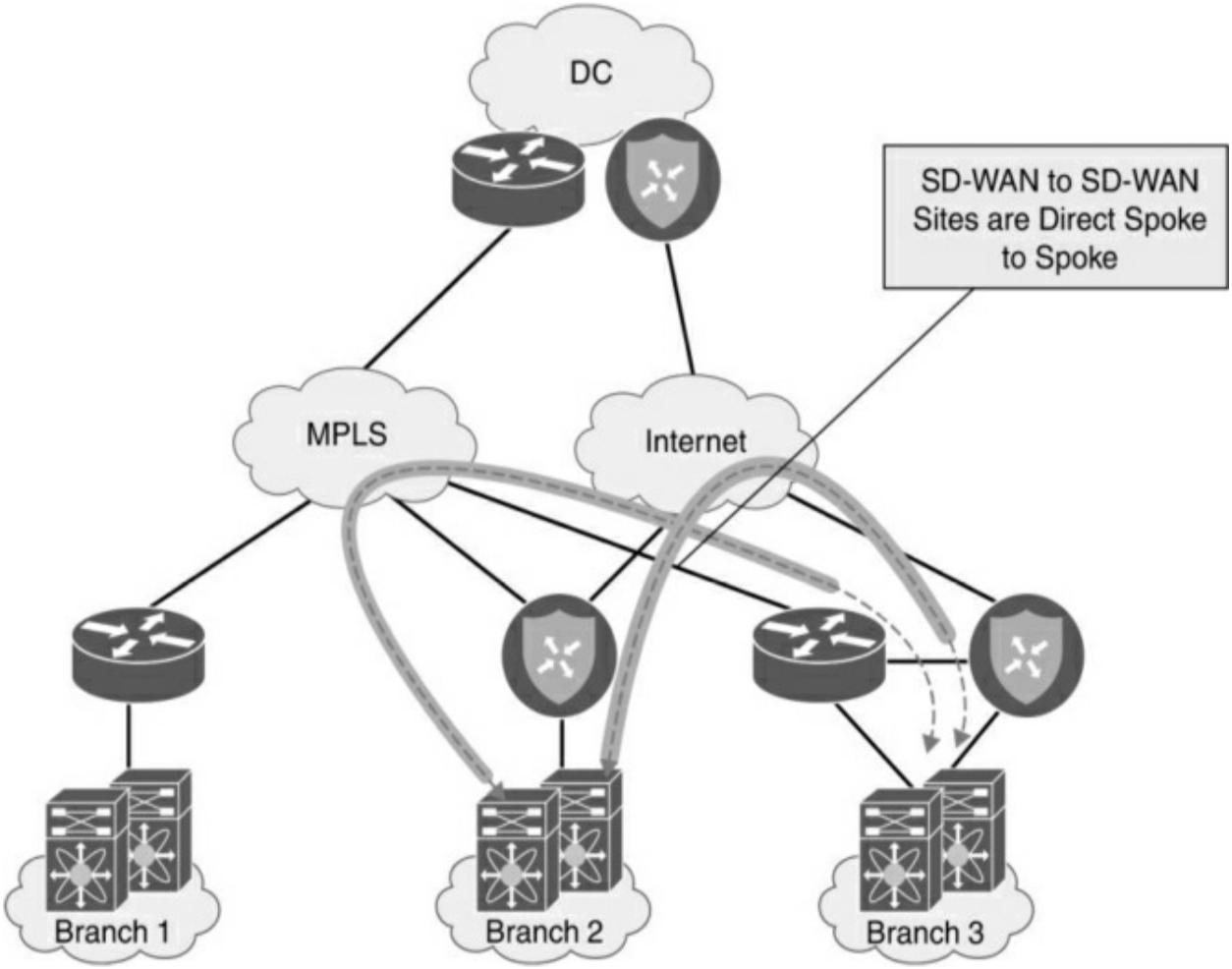


Figure 5-48 *Full Overlay with Underlay Routing: SD-WAN Site to SD-WAN Site Traffic Flow*

Note

In most cases, configuring full overlay and underlay integration at every site will add a significant amount of routing complexity to the network and, in some environments, can be a challenge to scale and control. Typically, voice applications can handle 300 ms of round-trip latency, and therefore this design might not even be required.

One way to implement full overlay and underlay integration at a branch is through the use of an existing MPLS CE router. This design is similar to the MPLS CE router integration solution discussed previously, with the difference being that the CE router continues to advertise the site prefixes into the MPLS underlay while the WAN Edge router simultaneously advertises the site prefixes into the overlay. Both the MPLS CE router and WAN Edge router advertise prefixes from the WAN into the LAN. However, the WAN Edge router should advertise SD-WAN migrated site prefixes (including data center prefixes) with a more attractive metric to retain traffic symmetry and to force Cisco SD-WAN sites to talk to each other via the overlay. While any routing protocol can be used to achieve this design, BGP is recommended for its native anti-transit logic. If any other routing protocol is used, tagging and filtering mechanisms should be used to avoid making the branch into a transit site. Figure 5-49 shows this design option.

Technet24

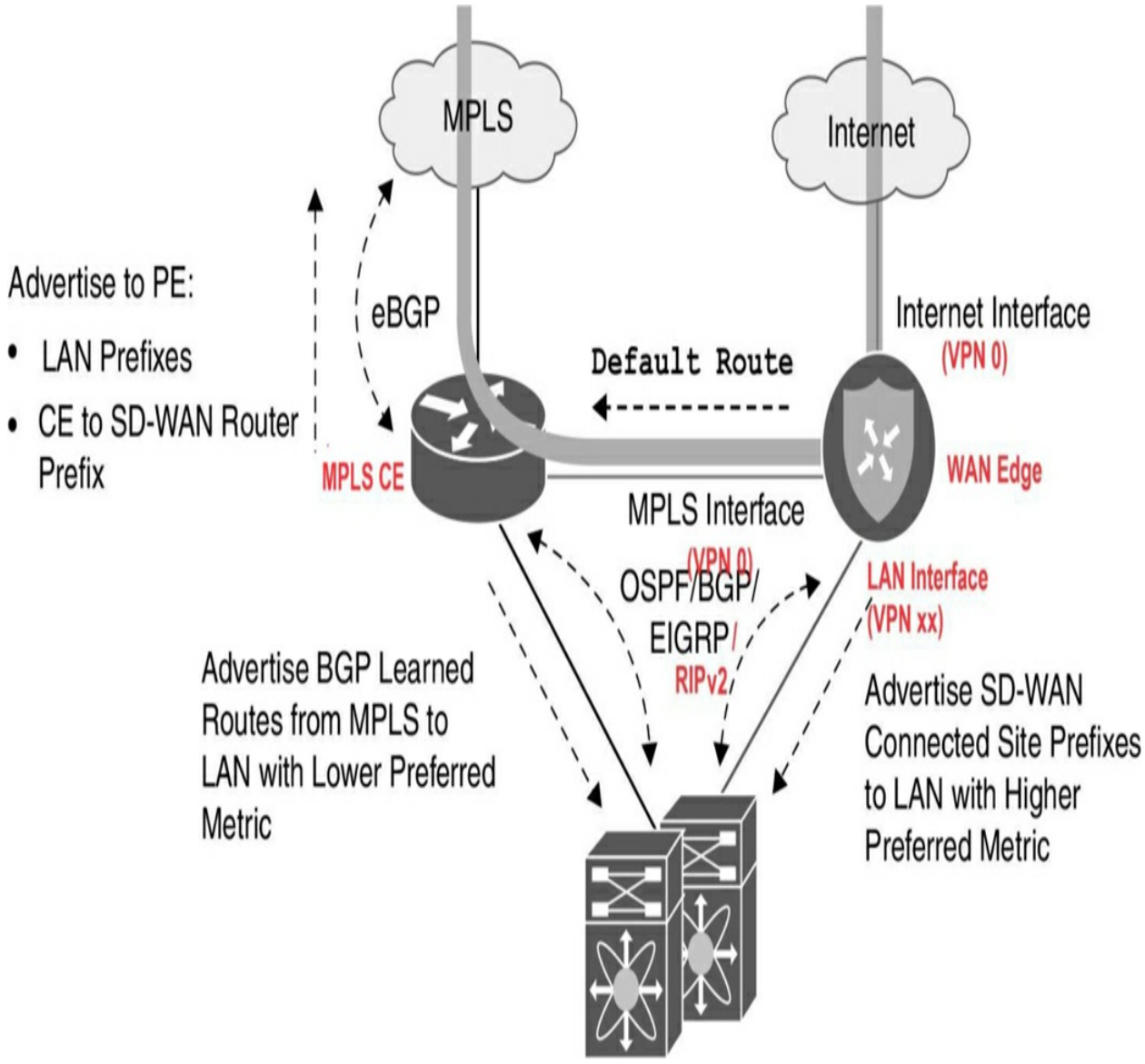


Figure 5-49 *Full Overlay with Underlay Routing: CE Integration*

Figure 5-50 shows the traffic flow for a branch with two transports. Traffic can flow directly through an Internet interface as well as through an interface that is connected to an existing MPLS CE router.

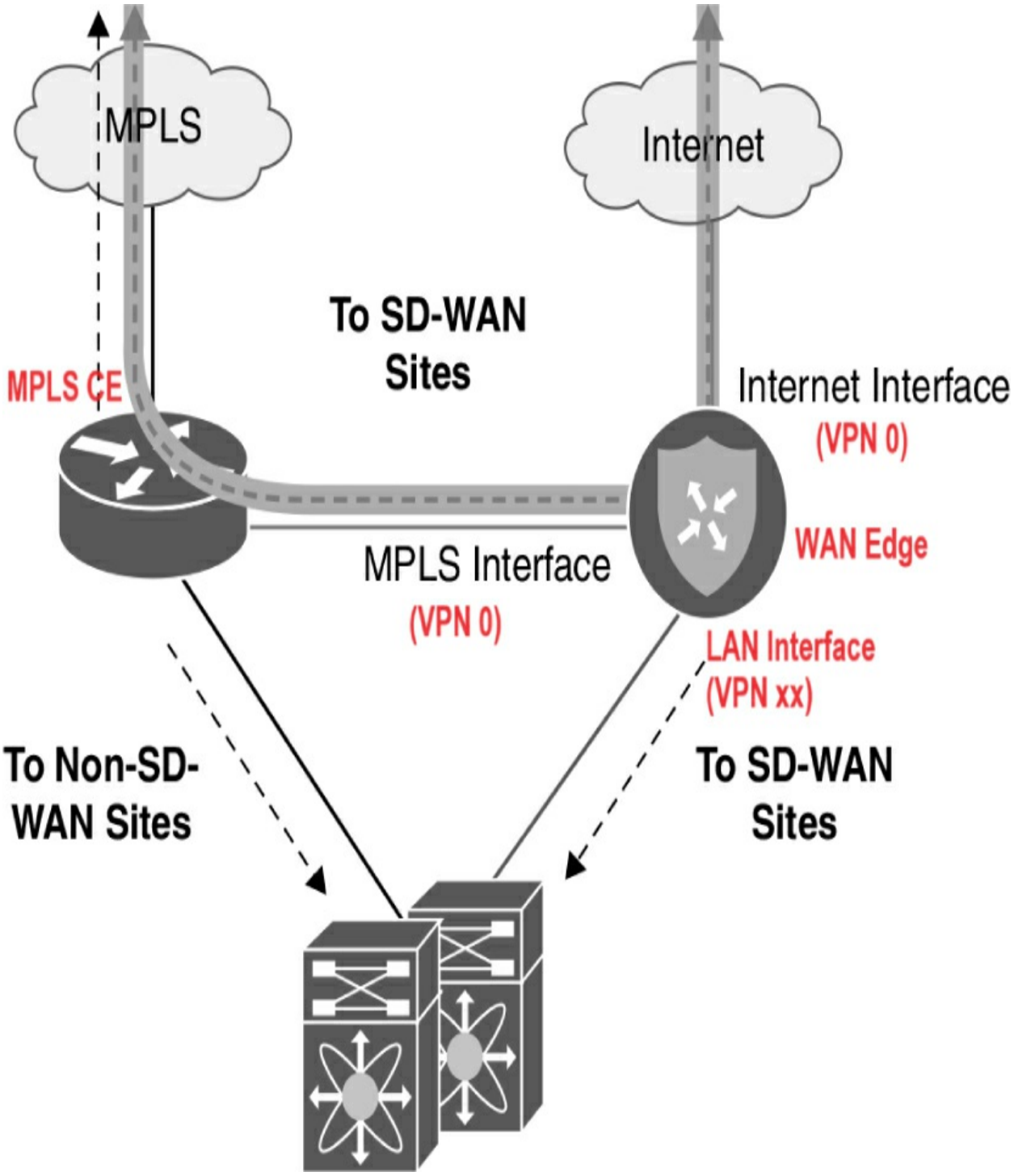


Figure 5-50 *Full Overlay with Underlay Routing: CE Integration Traffic Flow*

Full overlay and underlay integration can also be achieved without a

Technet24

dedicated MPLS CE router when only a WAN Edge router exists at the branch. The same principles apply as if there were a separate MPLS CE router; however, TLOC termination on the WAN Edge router is a bit different. Instead of configuring the TLOC on the physical interface connected to the MPLS carrier, a loopback interface is created in VPN 0, bound to the physical WAN interface, and configured as the TLOC. This configuration is required to remove TLOC configuration, along with the associated implicit ACL, from the physical interface. Without it, non-SD-WAN transit traffic received on MPLS from legacy sites will be dropped.

A routing protocol is configured between the WAN Edge router and the carrier in order to advertise the loopback interface IP address for control and data plane tunnel termination. A VPN 0 interface is then configured and connected to a downstream (LAN-side) Layer 3 switch. Finally, a routing protocol can be configured on this transit link to learn and advertise overlay routes. This configuration, in effect, creates a route leak between VPNs. Figure 5-51 shows this design option.

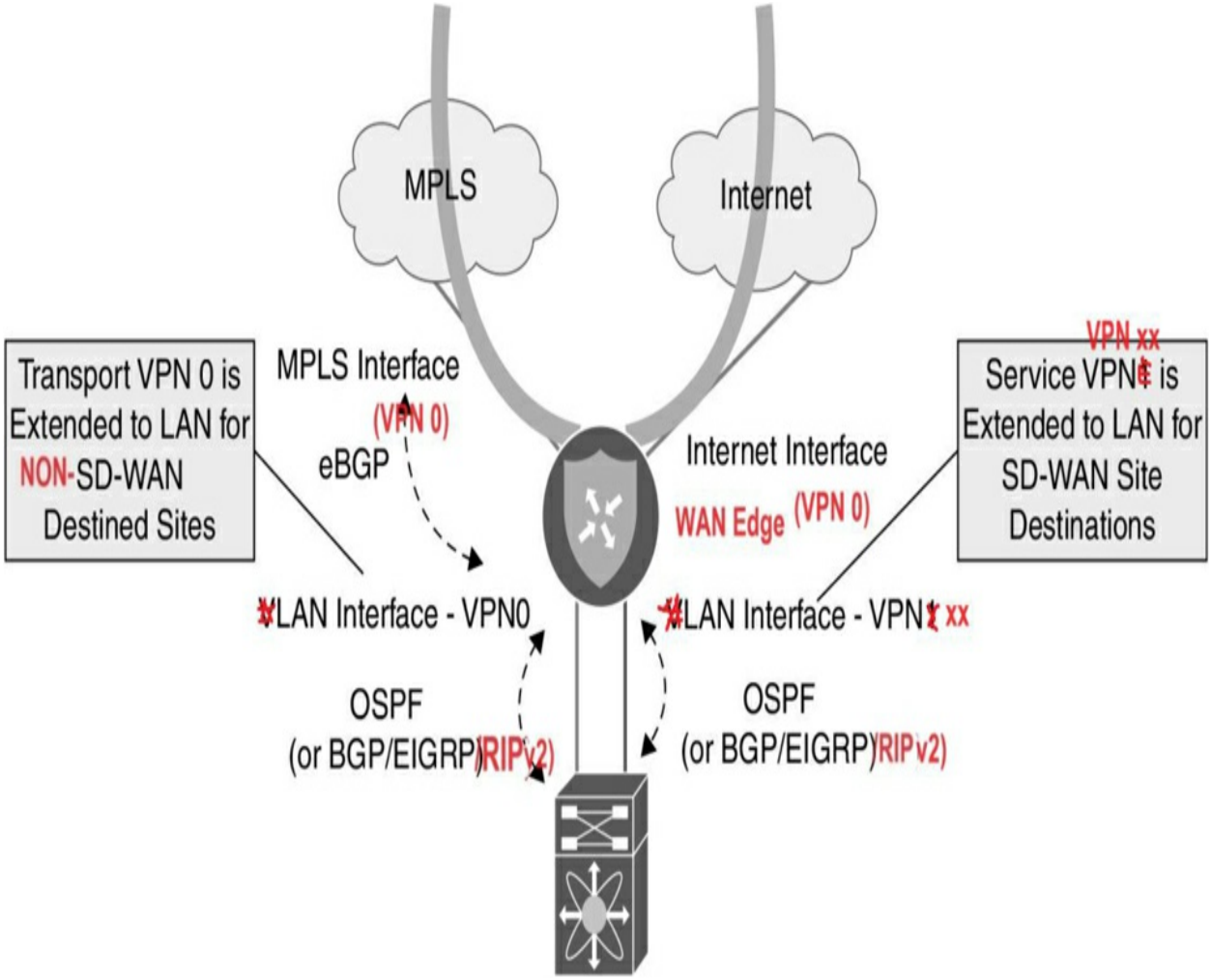


Figure 5-51 *Full Overlay with Underlay Routing: Without CE Integration*

Figure 5-52 illustrates the traffic flow for a site location with no integration with a CE device.

Technet24

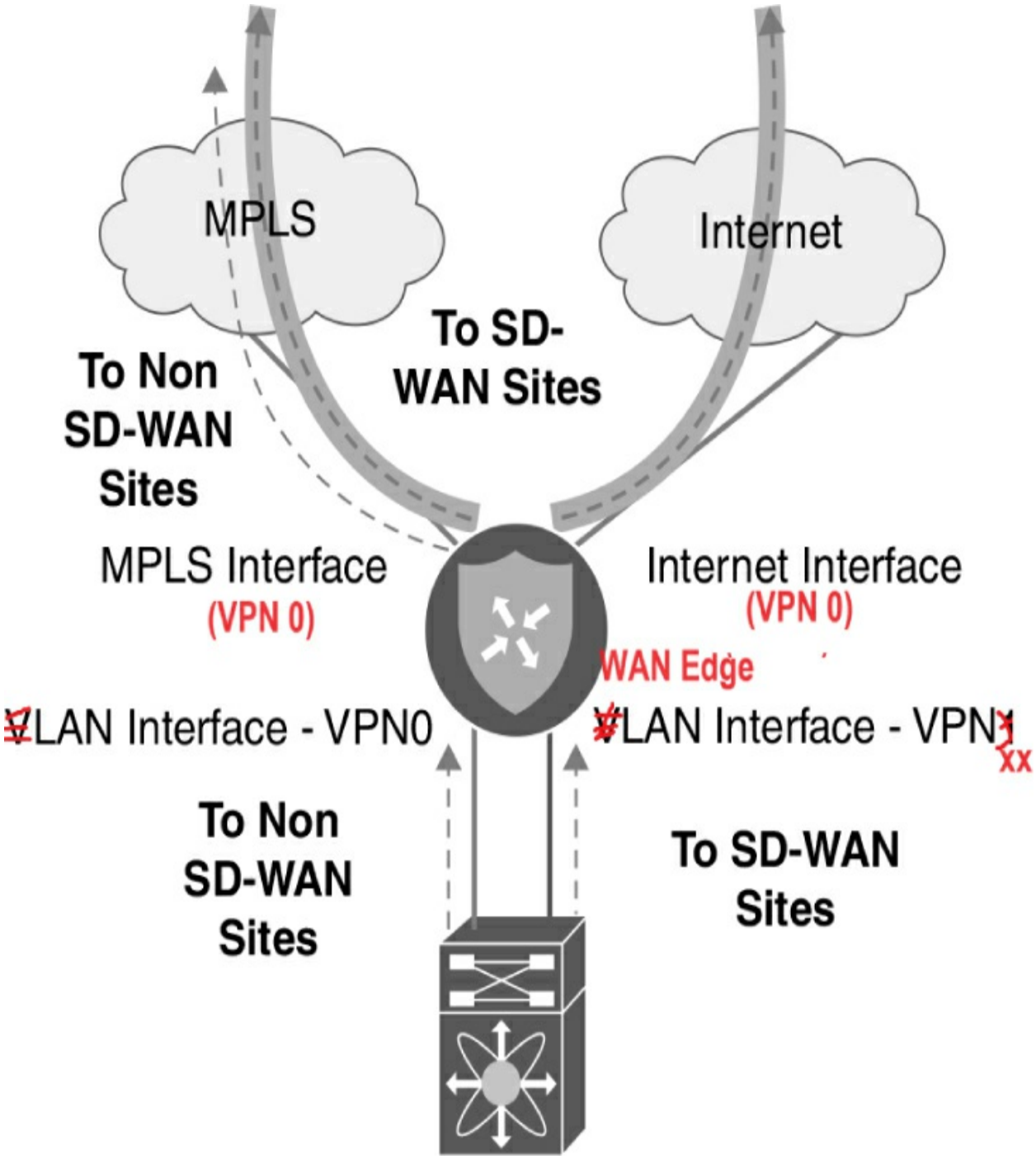


Figure 5-52 *Full Overlay with Underlay Routing: Without CE Integration Traffic Flow*

Note

Additional tagging, filtering, and best path manipulation may be required at both the data center and the remote sites participating in full overlay/underlay connectivity, depending on how routing between the overlay and underlay is accomplished. From a routing perspective, there are many valid ways to achieve full overlay and underlay integration. Each of these options deserves special attention to avoid routing loops and suboptimal routing. Every environment has unique complexities and caveats, so it’s important to think through all the different possible scenarios specific to your organization’s network that could affect the flow of traffic during and after migration.

Summary

This chapter covers the design methodology recommended for the implementation of Cisco SD-WAN. It discusses the importance of migration preparation, validated data center and branch designs, and overlay and underlay integration techniques. Cisco SD-WAN implementation requires a solid discovery and design period in which ample time is spent understanding the existing network thoroughly while also thinking about its future state. Preparation prior to deploying the Cisco SD-WAN network is key in ensuring that data center and branch cutovers are executed flawlessly. Understanding the benefits and caveats of all the supported data center and branch designs

Technet24

allows a network architect to select a design that meets the requirements of the business and provides additional Cisco SD-WAN functionalities while maintaining resiliency and performance. Every network has some degree of complexity, and the goal is to implement Cisco SD-WAN gracefully without increasing the complexity.

Review All Key Topics

Review the most important topics in the chapter, noted with the Key Topic icon in the outer margin of the page. Table 5-4 lists these key topics and the page number on which each is found.

Table 5-4 *Key Topics*

Key Topic Element	Description	Page
Section	Cisco SD-WAN design methodology	
Section	Cisco SD-WAN Control Components design	
Section	Cisco SD-WAN Control Components design	
Section	Cisco SD-WAN Multi-Region fabric	
Section	Loopback TLOC design	
Section	TLOC extensions	
Section	Integrating Cisco SD-WAN with existing networks	

Chapter Review Questions

1. What are some of main use case for Cisco Catalyst SD-WAN? (Choose three.)

a. Application performance optimization

Technet24

b. Secure direct Internet access

c. Multicloud connectivity

d. Data center traffic segmentation

2. Cisco SD-WAN simplifies connectivity to what kind of resources in a cloud? (Choose two.)

a. NaaS

b. IaaS

c. PaaS

d. SaaS

3. Which of the following is not a consideration for Cisco SD-WAN design?

a. Deployment scale

b. Wireless authentication

c. High availability and disaster recovery

d. VPN segmentation

4. For what region in a Multi-Region Fabric do you need to add a topology

control policy to ensure that traffic flow from one access region to another follows a certain preferred path?

a. None

b. Core region

c. Source access region

d. Destination access region

5. In which deployment model of Cisco SD-WAN Control Components does a customer assume the most responsibility?

a. Cisco-hosted cloud

b. MSP-hosted cloud

c. On-premises private cloud

d. Customer’s public cloud

6. Which Cisco SD-WAN component does not support high availability configuration?

a. Cisco Catalyst SD-WAN Analytics

b. Cisco Catalyst SD-WAN Manager

Technet24

c. Cisco Catalyst SD-WAN Controller

d. Cisco Catalyst SD-WAN Validator

7. What is the recommended disaster recovery mechanism for SD-WAN Manager?

a. Manual

b. Scheduled

c. Administrator-triggered failover

d. Arbitrator-triggered failover

8. How many SD-WAN Controllers are recommended for a deployment with 500 WAN Edge routers?

a. 0 as this is an optional component

b. 1 because 1 SD-WAN Controller has more than enough capacity for such a network

c. 2 for the redundancy

d. 3 because a clustered implementation requires an uneven number of devices

9. Which statement is true about SD-WAN Validator?

a. Only two SD-WAN Validators can be provisioned in an SD-WAN network.

b. It tracks the state of all the other SD-WAN Validators in the network.

c. It can be entered in the WAN Edge configuration either as an IP address or as a DNS name.

d. It runs the same virtual machine image as SD-WAN Controller, and the role is activated via the **system vbond** command with the **local** option.

10. True or false: Most SD-WAN deployments are done in a greenfield environment.

a. True

b. False

11. True or false: Migration to SD-WAN is a hard cut, with all data centers and remote sites migrated simultaneously.

a. True

b. False

12. What are some of the most important preparations that should be made

Technet24

prior to migration? (Choose two.)

a. Reloading all routers

b. Ensuring that every router has a maintenance contract

c. Analyzing the existing topology, routing, and traffic engineering

d. Cisco SD-WAN policy design and configuration

13. What type of SD-WAN-specific configuration values should be defined prior to migration? (Choose three.)

a. Site IDs

b. VPN IDs

c. OSPF hello and dead timers

d. BGP MED values

e. TLOC colors

14. True or false: Standing up WAN Edge routers alongside the current WAN infrastructure is the preferred method for Cisco SD-WAN data center integration.

a. True

b. False

15. From a design perspective, how can you improve latency when migrating to Cisco SD-WAN while transiting hubs?

a. Enable TCP optimization.

b. Move the sites closer to each other.

c. Designate multiple regional hubs as transit points.

d. Use dedicated Layer 2 circuits.

16. True or false: The most common way to integrate transport-side connections on a WAN Edge router into the network is by sharing a single interface in VPN 0 for all carriers/transports.

a. True

b. False

17. What is the difference between bind mode and unbind mode when configuring a loopback TLOC?

a. Bind mode forms control connections, and unbind mode does not.

b. Unbind mode takes less router CPU than bind mode.

Technet24

c. Unbind mode allows for public and private color connectivity, whereas bind mode does not.

d. Bind mode ensures that traffic destined to the loopback will be carried to and from the mapped physical interface. Unbind mode does not have this behavior.

18. What are three different Cisco SD-WAN branch migration options? (Choose three.)

a. Complete CE replacement with a single WAN Edge router

b. WAN Edge router running inline bridge mode

c. Integration with an existing CE router

d. Complete CE replacement with dual WAN Edge routers

19. True or false: A TLOC extension can only be configured for private colors.

a. True

b. False

20. Which routing protocols are supported for service-side integration with the LAN? (Choose three.)

a. OMP

b. OSPF

c. ISIS

d. EIGRP

e. ODR

f. eBGP

21. What important consideration needs to be made at the data center when implementing integration with existing networks?

a. Only OSPF should be used.

b. ECMP should be configured.

c. WAN Edge routers should be running the latest code.

d. Filtering toward the WAN Edge router should be configured.

References

"Cisco SD-WAN Design Guide," https://www.cisco.com/c/en/us/td/docs/solutions/CVD/SDWAN/cisco-sdwan-design-guide.html.

Technet24

“Cisco SD-WAN Large Global WAN Design Case Study,” https://www.cisco.com/c/en/us/td/docs/solutions/CVD/SDWAN/Cisco_SDW

“Cisco Catalyst SD-WAN High Availability Configuration Guide, Cisco IOS XE Catalyst SD-WAN Release 17.x,” https://www.cisco.com/c/en/us/td/docs/routers/sdwan/configuration/ha-scaling/ios-xe-17/high-availability-book-xe/m-high-availability-and-scaling.html.

“Cisco Recommended SD-WAN Software Versions for Controllers and WAN Edge Routers,” https://www.cisco.com/c/en/us/support/docs/routers/sd-wan/215676-cisco-tac-and-bu-recommended-sd-wan-soft.htm

Chapter 6. Introduction to Cisco SD-WAN Policies [This content is currently in development.]

This content is currently in development.

Technet24

Chapter 7. Centralized Control Policies

This chapter covers the following topics:

• **Centralized Control Policy Overview:** This section discusses the basics of centralized control policies and the directionality of policies when applied to a SD-WAN Controller.

• **Use Case 1: Isolating Remote Branches from Each Other:** This section covers the process of building a version of a hub-and-spoke network by applying a centralized control policy to limit the data plane tunnels that are built in the SD-WAN fabric.

• **Use Case 2: Enabling Branch-to-Branch Communication Through Data Centers:** This section covers the process of building another version of a hub-and-spoke network, using summarization and TLOC lists to enable sites that do not have direct data plane connectivity to communicate.

• **Use Case 3: Traffic Engineering at Sites with Multiple Routers:** This section covers the use of the TLOC Preference attribute to manipulate how traffic flows into sites with more than one WAN Edge router to influence symmetric routing.

• **Use Case 4: Preferring Regional Data Centers for Internet Access:** This section covers the use of the OMP Route Preference attribute to perform traffic engineering on a per-prefix basis.

• **Use Case 5: Regional Mesh Networks:** This section discusses how to create subsets of the fabric that have a full mesh of data plane connectivity, even while the whole fabric does not.

• **Use Case 6: Enforcing Security Perimeters with Service Insertion:** This section covers the use of service insertion to be able to direct a flow from anywhere in the fabric to a local or remote service.

• **Use Case 7: Isolating Guest Users from the Corporate WAN:** This section covers the use of VPN membership policies to be able to restrict which VPNs can join the overlay fabric.

• **Use Case 8: Creating Different Network Topologies for Each Segment:** This section covers the use of control policies to create multiple, arbitrary topologies that can be applied on a per-VPN basis.

• **Use Case 9: Creating Extranets and Access to Shared Services:** This section covers the construction of extranets to enable connectivity with business partners while still maximizing security posture.

Network administrators can use centralized control policies to manipulate the way traffic flows throughout the Cisco SD-WAN fabric. Fundamentally, centralized control policies are the mechanism through which the control plane information that is advertised by Overlay Management Protocol (OMP) between the SD-WAN Controller (formerly vSmart) and WAN Edge routers is manipulated and/or filtered. By manipulating or filtering this information

Technet24

in the control plane, network administrators can influence the way that end-user traffic is forwarded in the data plane in order to accomplish their business objectives. This chapter explores several common business-relevant use cases, the network designs necessary to achieve them, and the centralized control policies used to implement them.

Centralized Control Policy Overview

Throughout the next several chapters, where we examine policies in Cisco SD-WAN, we will be using the topology illustrated in Figure 7-1.

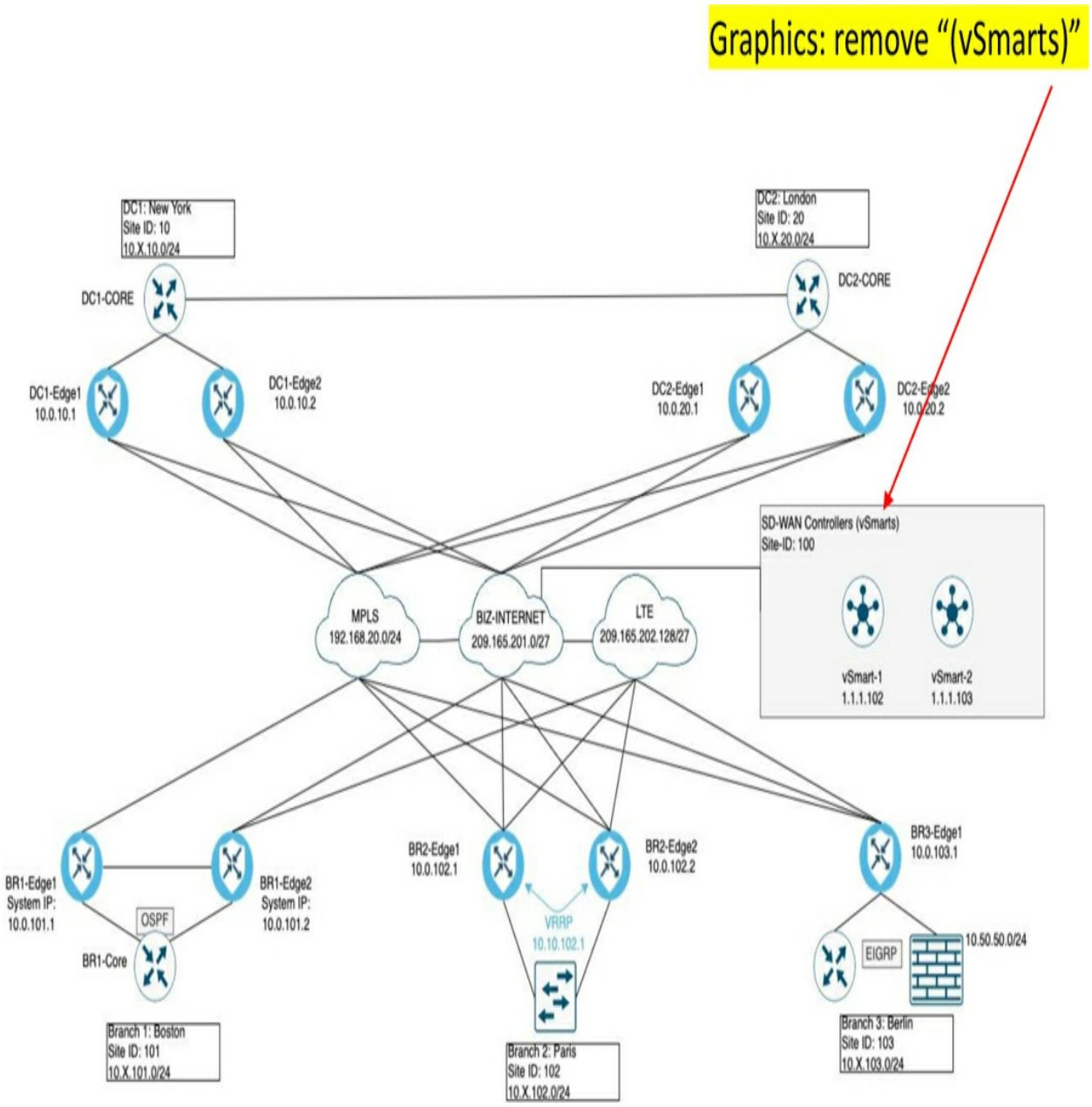


Figure 7-1 *Network Topology Overview*

This simple network topology features two different data centers (DCs) and three branch sites. Each data center has both an MPLS transport and an Internet transport. Each branch, in addition to having an MPLS transport and an Internet transport, also has an LTE transport. The host names, site IDs, router IDs, and network prefixes are also shown in this figure. The service-

Technet24

side addressing in this network follows the 10.X.Y.0/24 structure, where X signifies the service-side VPN, and Y signifies the site ID. For the first few use cases in this chapter, we will be focusing on VPN 10, but as we progress through the use cases, we will use additional VPNs 20 and 50. Therefore, all of the service-side addressing will be in the 10.X.Y.0/24 addressing blocks. The SD-WAN controllers and their system IP addresses are also indicated in this diagram. This topology will allow us to explore different types of policies, how they interact with the SD-WAN fabric, and how network administrators can apply these policies to solve business problems.

As discussed in Chapter 6, "Introduction to Cisco Catalyst SD-WAN Policies," each type of policy has a specific directionality to it. In the case of centralized control policies, policies can be applied in either the inbound or the outbound direction. This directionality is always from the perspective of the SD-WAN Controller, as shown in Figure 7-2.

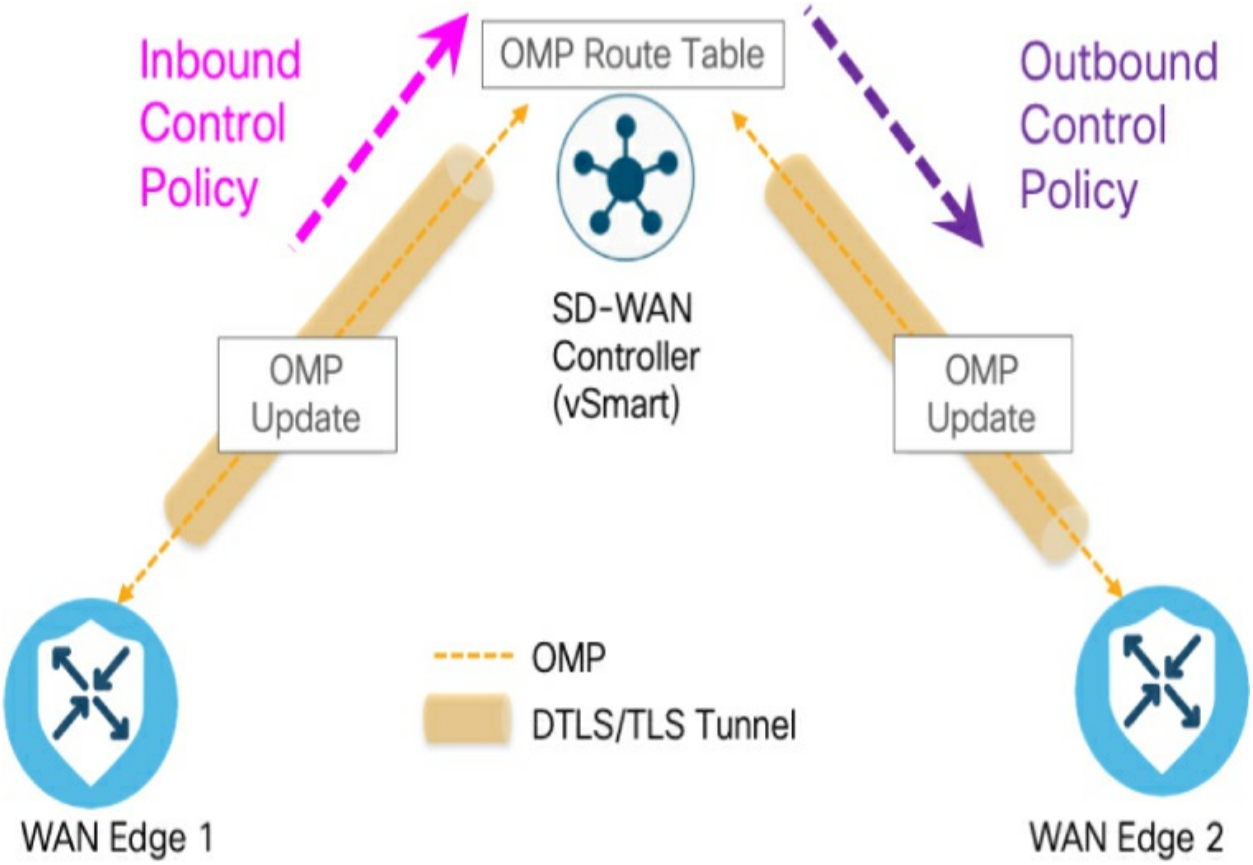


Figure 7-2 *Inbound and Outbound Control Policies*

Key Topic

Inbound policies are applied inbound on the SD-WAN Controllers on updates from the sites configured on the site list before the routes are processed through the best-path selection algorithm and before the routes are inserted into the OMP table on the SD-WAN Controllers. Therefore, any manipulation by inbound control policies will be evident in the SD-WAN Controller best-path selection process and, in turn, will be evident in the OMP advertisements made to all the other WAN Edge routers. Conversely, policies that are applied outbound apply after the SD-WAN Controller best-path selection process has been completed and are limited in scope to only

Technet24

those site IDs listed in the control policy application configuration. Centralized control policies that are applied inbound tend to be much more global in nature, whereas outbound control policies can be much more limited and targeted in scope and application.

In the following sections, we examine several different sets of business requirements and how network administrators can use centralized control policies to solve for these use cases. These use cases are meant to address common applications of centralized control policies as well as to provide an illustrative review of many of the building blocks of centralized control policies from which network administrators can build policies to accomplish their objectives.

Note

This chapter is based on SD-WAN Version 20.12.

Use Case 1: Isolating Remote Branches from Each Other

The first use case we will be exploring involves turning the topology of the fabric from a full mesh into a hub-and-spoke design, with the data centers as hubs and the branch offices as spokes. Network designs like this are often used for several reasons: In many types of networks where security is a concern—like retail and finance, as well as a couple of market vertical examples—there is little need to be able to communicate from one branch

directly to another branch. In such networks, the structure of the network can be changed to reflect the desired traffic flow, and the business intent can be encoded directly into the SD-WAN fabric. Furthermore, if this is a large network consisting of hundreds or thousands of spokes, the devices deployed at the branch sites may not have the capacity to build tunnels to all of the other branch offices. In this way, the devices at the branch offices do not need to be sized to handle hundreds or thousands of IPsec tunnels.

As discussed in Chapter 3, "Control Plane and Data Plane Operations," the default state of the Cisco SD-WAN fabric with no policies applied is a full mesh. That is, every WAN Edge router builds an IPsec tunnel to every other router that it has reachability to. In our sample network, tunnels from the MPLS interfaces will only be built to the other MPLS interfaces. By contrast, tunnels from BIZ-INTERNET will be built to both BIZ-INTERNET and LTE interfaces on other routers. Tunnels from the LTE color will be built to both the BIZ-INTERNET and LTE interfaces on the other routers in the fabric. In this relatively simple network of only five sites and only nine routers, each of the WAN Edge devices at Branch 2 will build 24 different tunnels, as shown in the real-time output in Figure 7-3.

Technet24

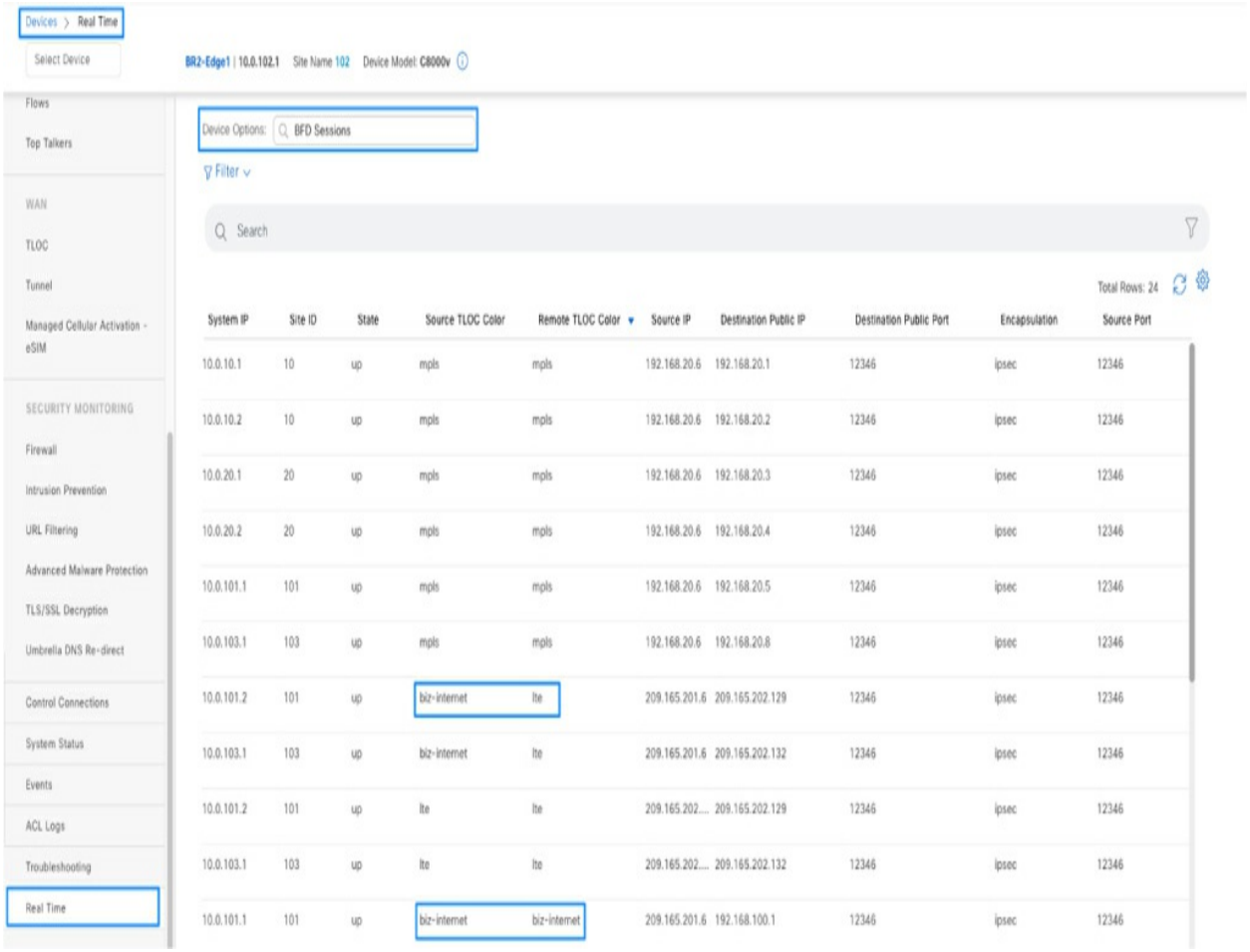


Figure 7-3 *BR2-Edge1 Tunnels with No Control Policy Applied*

Note

The number of BFD sessions is directly influenced by the **restrict** attribute that is configured on the tunnel interfaces with the MPLS color. As outlined in the "Key Settings in Device Templates and Configuration Groups" section of Chapter 4, "Onboarding and Provisioning," this is a relevant setting for preventing session establishment between different type of transports.

You can access the real-time output, as shown in Figure 7-3, in SD-WAN Manager (formerly vManage) GUI by selecting Monitor > Devices > [*Device*]. You can use the Monitor > Devices display to track lots of information that administrators will find useful throughout the provisioning and operation of the Catalyst SD-WAN fabric. The Real Time option can be found at the very bottom of the list of elements. Once this option element is selected, the Device Options field at the top of the page can be used to query the device and display the output of a **show** command that would be used from the CLI interface. This feature is named Real Time because it queries the device *in real time* and does not rely on cached data, as do many of the other elements in the Monitor section of SD-WAN Manager. With this option selected, you might occasionally notice a slight delay as SD-WAN Manager queries the device for the requested real-time output.



As discussed in Chapter 3, the process of building data plane tunnels is controlled by the advertisement of Transport Location (TLOC) identifiers. If a WAN Edge device receives a TLOC route from the SD-WAN Controller, it will attempt to build a data plane tunnel to that TLOC and establish an encrypted Bidirectional Forwarding Detection (BFD) session, as long as the TLOC attributes such as **restrict** or **tunnel-group** are set to their default values. Monitoring BFD sessions, as shown in Figure 7-3, is a good way to understand whether tunnels are trying to be formed and whether they have

Technet24

been formed successfully. To change this behavior in the data plane, you must filter which TLOC routes are advertised from the SD-WAN Controller to the WAN Edge routers.



If the BFD session is working as expected, and there is bidirectional data plane connectivity, the BFD session will be listed in the “up” state. If the TLOC advertisement was received from the SD-WAN Controller, but the data plane tunnel is unable to form correctly, the BFD session will be listed in the “down” state. If the data plane tunnel does not appear in the list of BFD sessions, the WAN Edge device is not attempting to build the data plane tunnel. This behavior is generally caused for one of two reasons: Either the WAN Edge device has not received the TLOC advertisement from the SD-WAN Controller or the data plane tunnel is prohibited from being built by either the **restrict** or **tunnel-group** settings.

The current state of the network can also be examined in terms of the ways that traffic flows through the network. In Example 7-1, you can see that from Branch 2, the destination address 10.10.103.1 (which resides at Branch 3) is reached directly.

Example 7-1 *Tracing from BR2-Edge1 to BR3-Edge1*

```
! Traceroute from BR2-Edge1 to BR3-Edge1:
```

```
! The traceroute is successful, and shows that
BR2-Edge1# traceroute vrf 10 10.10.103.1
<<<Output omitted for brevity>>>
  1 LTE-BR3-Edge1 (209.165.202.132) 32 msec 1 r
  2 BR3-Edge1-Core (172.16.103.2 207) msec 1 ms
```

```
! Traceroute from BR2-Edge1 to DC1 services:
! The traceroute is successful; resources in th
BR2-Edge1# traceroute vrf 10 10.10.10.100
<<<Output omitted for brevity>>>
  1 BIZ-INET-DC1-Edge1 (209.165.201.1) 37 msec
  2 DC1-Edge1-Core (172.16.10.2) 22 msec 1 msec
  3 DC1-Service (10.10.10.100) 128 msec 1 msec
```

In addition to the prefixes at Branch 3, Example 7-1 also shows that the prefixes in the data center are directly reachable. This is also reflected in the routing table, as shown with the real-time output, where the prefix 10.10.103.0/24 is reachable with the next-hop system IP address 10.0.103.1 (which is the system IP address of BR3-Edge1, as shown in Figure 7-1), and prefix 10.10.10.0/24 is reachable through the next-hop system IP address 10.0.10.2 (which is the system IP address of DC1-Edge2), as shown in Figure 7-4.

Technet24

Note

Remember that a TLOC identifier is composed of system IP address, color, encapsulation, and preference settings. The system IP address is independent of any IP address configured on device interfaces.

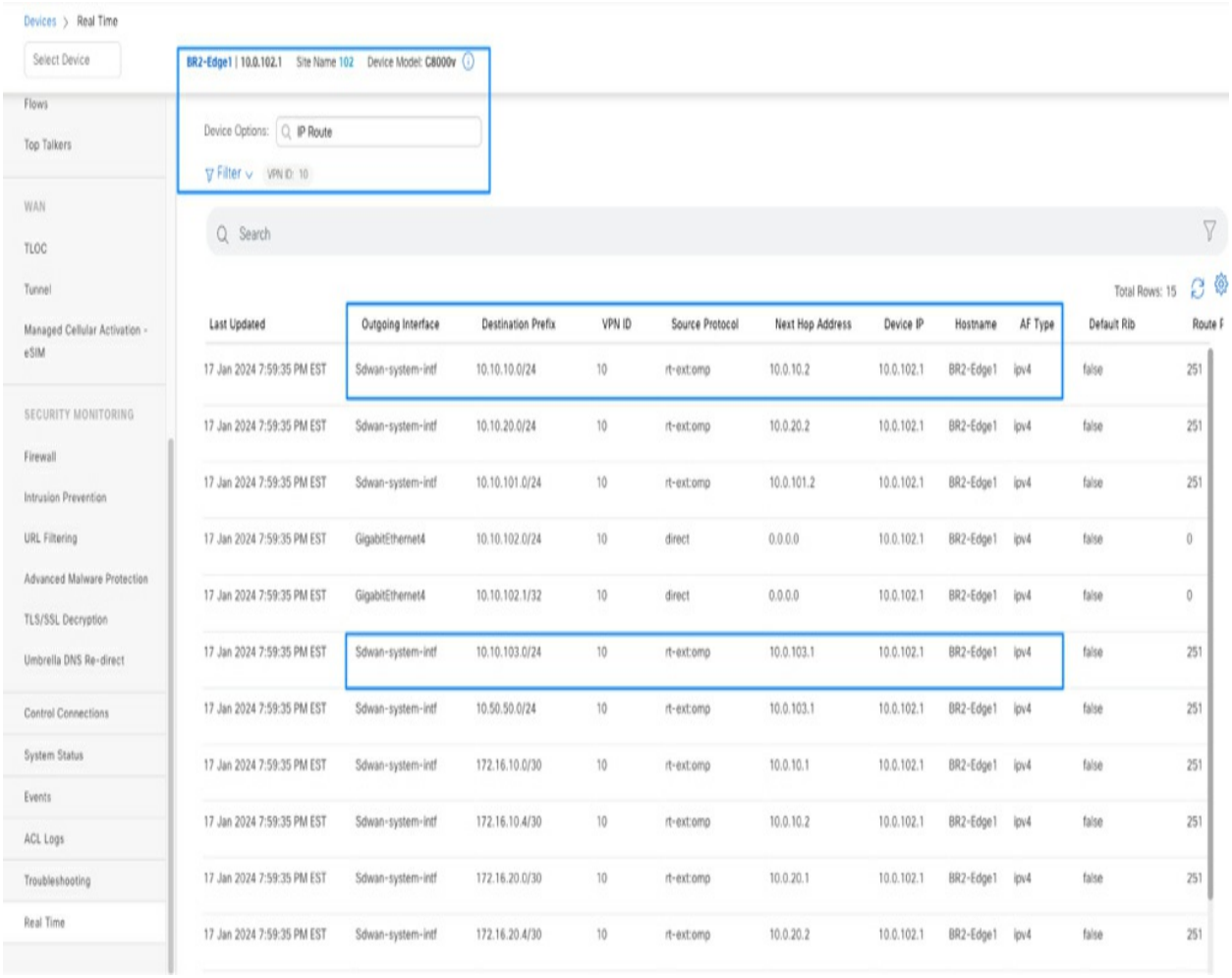


Figure 7-4 *Routing Table for VPN 10 on BR2-Edge1 with No Centralized Policy Applied*

In order to implement the business intent to eliminate branch-to-branch communication and to reduce the number of tunnels that are established, a centralized policy can be created and applied so that the branches will only build tunnels with the data center WAN Edge devices. As each WAN Edge router will attempt to build tunnels to all of the TLOCs that it receives from the SD-WAN Controller, such a policy is architected to restrict the TLOC routes that are advertised to the branches to be only the TLOCs from the data centers (where the branches should still build their control connections).

The advertisement of TLOC routes is depicted in Figure 7-5 and involves the following steps:

All of the WAN Edge routers advertise their local TLOC routes to the SD-WAN Controllers.

The SD-WAN Controllers advertise all of the received TLOC routes back to all the WAN Edge devices. For the data center WAN Edge device, this means that the router receives all of the TLOC routes in the SD-WAN Controller table, including a copy of its own TLOCs. An outbound control policy is applied to SD-WAN Controllers that restricts the advertisement of the TLOC to only T1 and T2 (learned from the data center) to Branch 1 and Branch 2.

Note

The advertisements that the data center WAN Edge devices receive in step 2 contain all of the TLOCs in the network {T1,

Technet24

T2, T3, T4, T5, T6}, which is different from the TLOC routes received by the branches. The outbound control policy has limited the TLOCs that are advertised to the branches to only {T1, T2}, the TLOCs from the data center.

The effect of this policy can be seen when the data plane tunnels are built: All of the WAN Edge routers build data plane tunnels to all of the TLOCs they have received from the SD-WAN Controllers. For a data center WAN Edge device, this means that tunnels will be built to T3, T4, T5, and T6, the TLOCs of the branch routers. For the branches, tunnels will only be built to T1 and T2, the data center.

Note

There is no tunnel built from T3 to T5 or from T4 to T6 in Figure 7-5. Since the WAN Edge router at Branch 1 never received an advertisement for T5 and T6, it is unaware of those TLOCs and will not attempt to build those tunnels. Manipulating and restricting the advertised TLOCs, as shown in Figure 7-5, is the fundamental way to control which data plane tunnels are built in the SD-WAN fabric, and it is therefore the primary way that network administrators control the structure of the fabric.

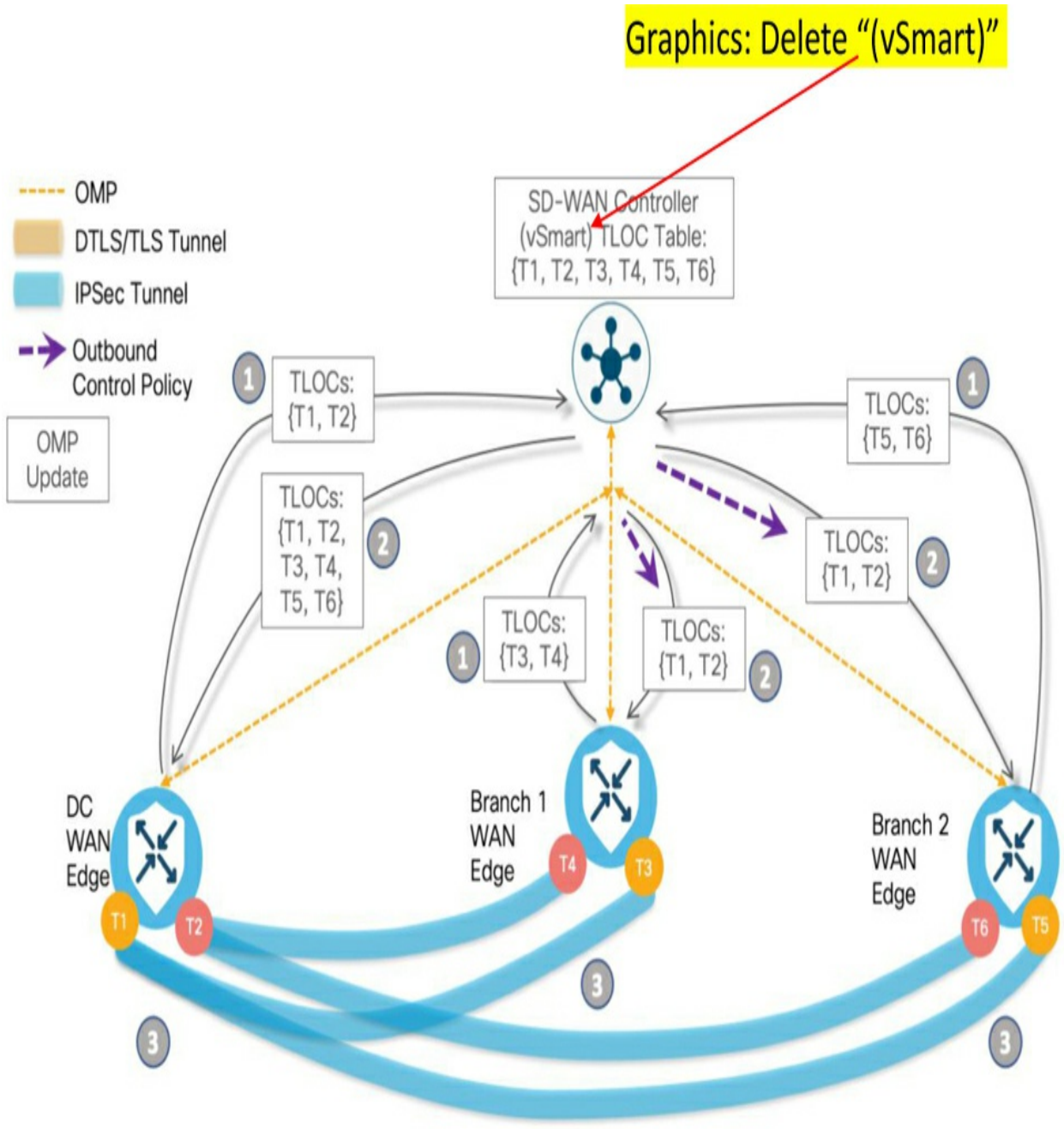


Figure 7-5 *Illustration of Control Policies Filter TLOCs*

Note

For the first two policy examples in this chapter, screenshots of the process to build and modify the necessary policies will

Technet24

be provided for both widely deployed traditional policies and a newly added topology approach. For brevity, the remaining policies in this chapter will be provided only with the CLI output to reflect the logic process, although the recommendation is to leverage the SD-WAN Manager GUI as it simplifies the process.

Custom Control Policy with a Traditional Workflow

The first step in constructing a custom control policy with a traditional workflow would be to open the Policy Wizard by clicking Add Policy from the Configuration > Policies screen in SD-WAN Manager, as shown in Figure 7-6.

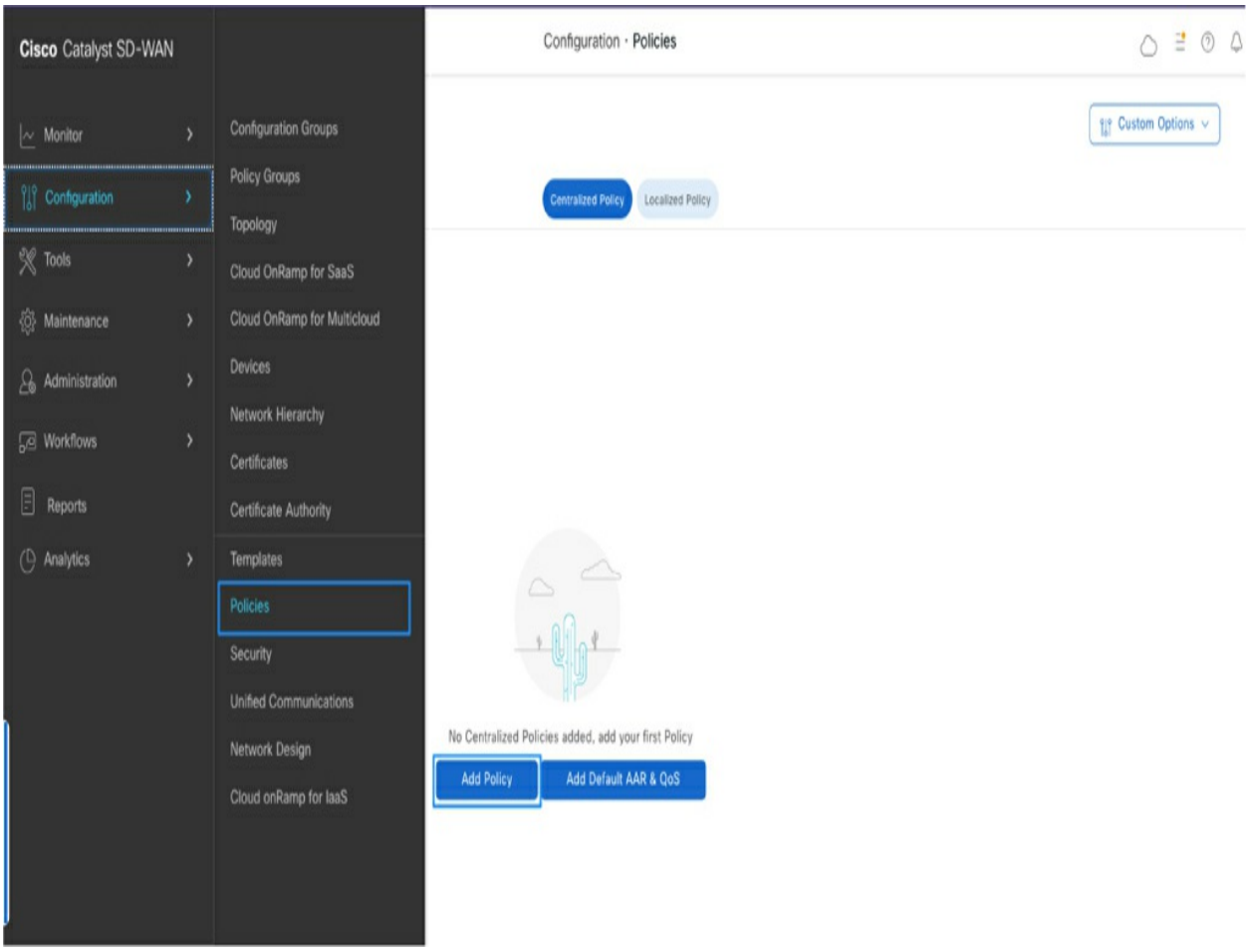


Figure 7-6 *Opening the Policy Wizard*

In the Policy Wizard, the first option is to configure all the criteria (lists) that will be used within this policy. For this first use case, you need to create two site lists: one that encompasses the site IDs of the data centers and a second that encompasses the site IDs of the branch offices. You create the site list by clicking on New Site List and typing in a friendly name and a specific site ID, a range of site IDs (separated with hyphens), or multiple site IDs (separated with commas). You can see these two lists in Figure 7-7. The ranges of site IDs configured in these lists reflect the ranges of site IDs identified in the network topology shown in Figure 7-1.

Technet24

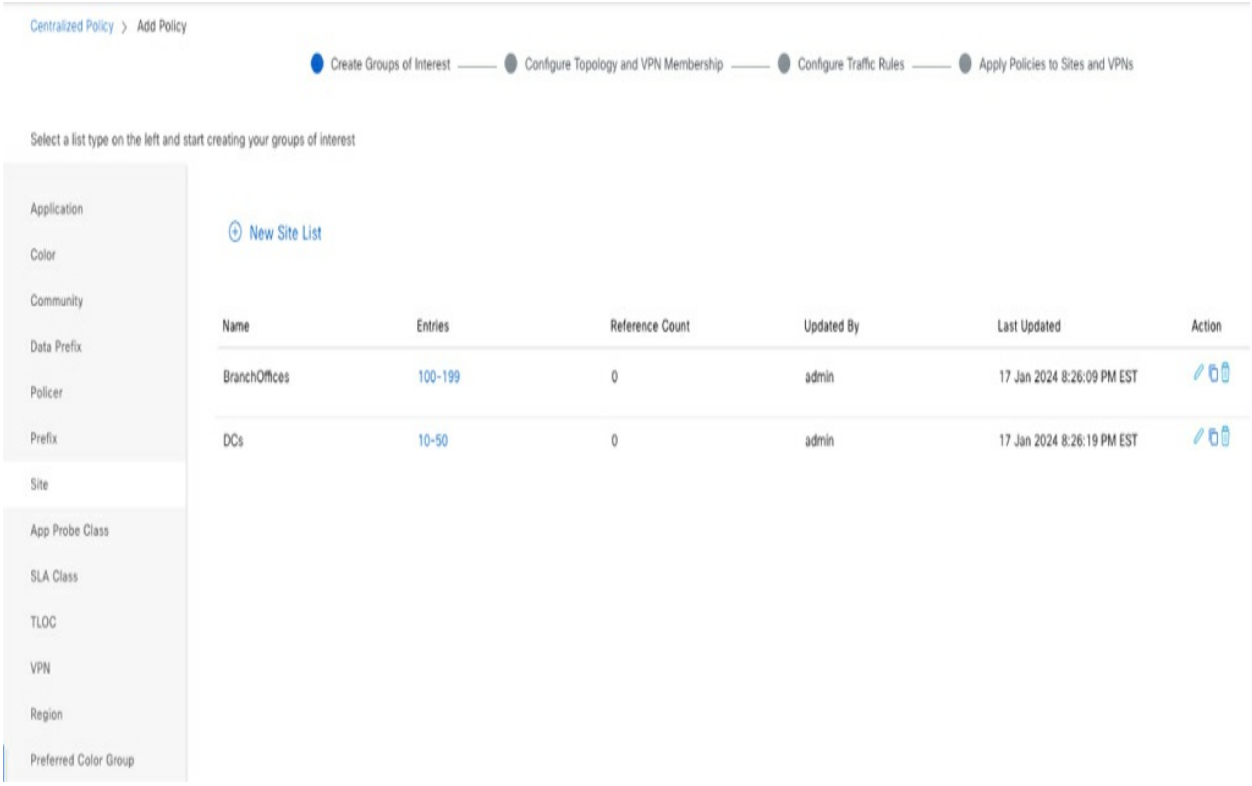


Figure 7-7 *Configuring Site Lists for Use in Centralized Policies*

After you click the Next button at the bottom of the Lists page, the wizard moves to the Configure Topology and VPN Membership page. On this page, click the Add Topology drop-down and select Custom Control (Route & TLOC) from the combo box, as shown in Figure 7-8.

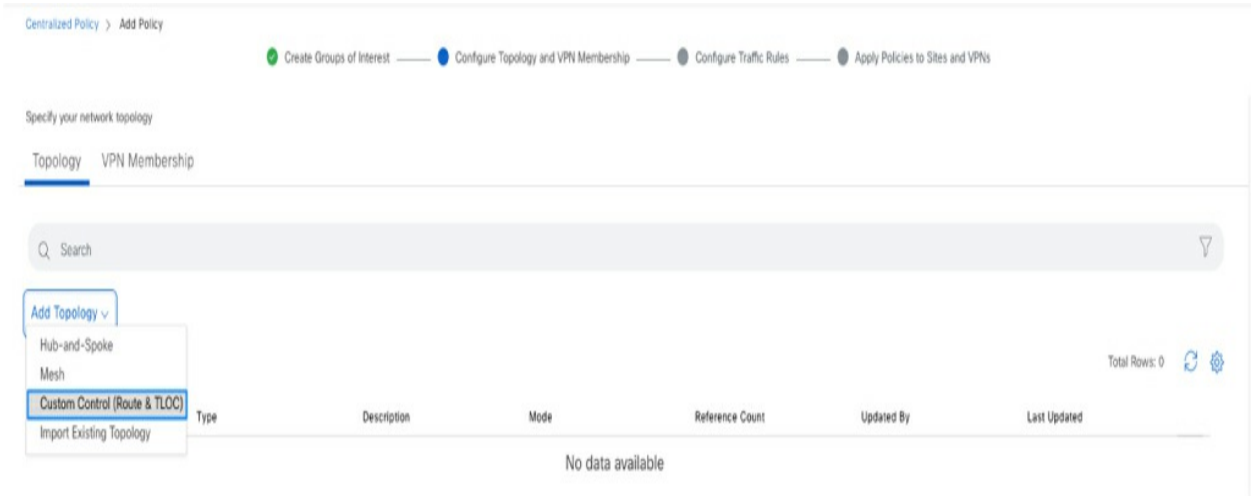


Figure 7-8 *Creating a Custom Control Policy*

Note

The SD-WAN Manager GUI includes wizards to build hub-and-spoke (where only hubs' routes are advertised) and mesh topologies, as shown in Figure 7-8. While some network administrators find these assisted workflows easier to use, this chapter focuses on custom control policies to better illustrate the construction of the policies and the principles behind their use.

The first pieces of configuration necessary for every control policy are the policy name and description, as shown in Figure 7-9.

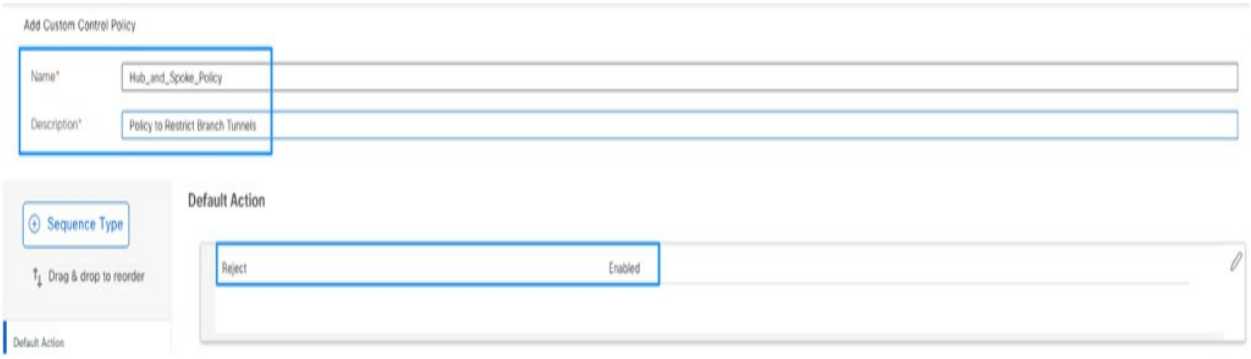


Figure 7-9 *Setting the Default Action in the Control Policy*

It is also important to notice in Figure 7-9 that the default action in custom route and TLOC policies is Reject, but this can be changed. If an OMP route or a TLOC is not matched by any other sequence in the policy and explicitly

Technet24

permitted, it will not be advertised from the SD-WAN Controller to the WAN Edge router.

Next, click the Sequence Type button and choose a TLOC sequence, as shown in Figure 7-10.

Next, create a new rule in the policy by selecting the + Sequence Rule option. In this sequence, specify a site list as the match criterion and match on the list called DCs, created previously. Finally, specify the action Accept with no further actions and then select Save Match and Actions, as shown in Figure 7-11.

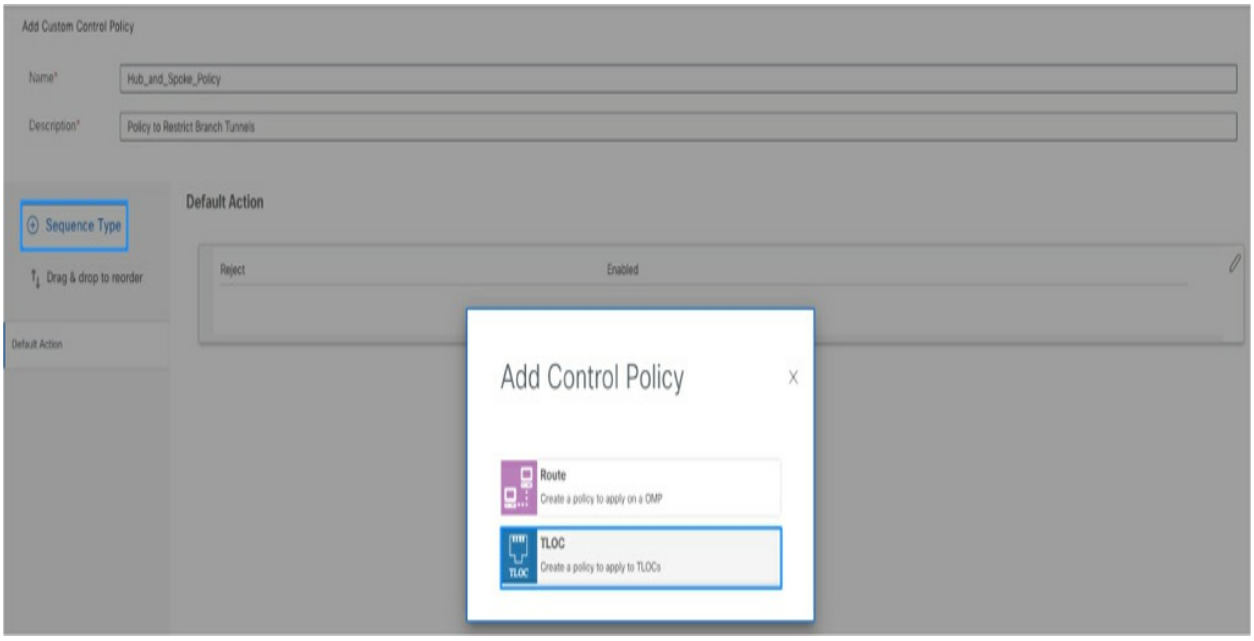


Figure 7-10 *Adding a TLOC Sequence to a Centralized Control Policy*

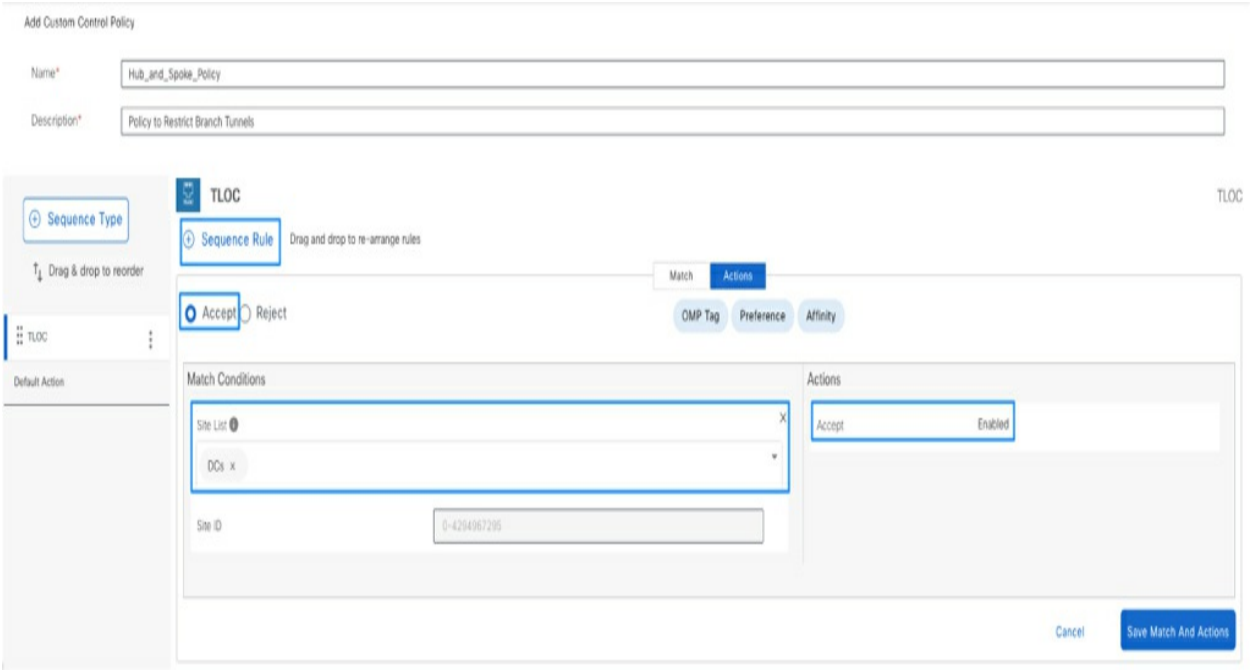


Figure 7-11 *Creating a TLOC Sequence Rule to Accept the DC TLOCs*

This TLOC sequence rule, in combination with the default rule that rejects anything that is not explicitly permitted, will result in only the DC TLOCs (that is, only the TLOCs from sites specified in the DCs site list) being advertised to the sites where this policy is applied. Once the TLOCs from the branch sites have been implicitly filtered, the network will have a topology analogous to that shown in Figure 7-5, and data plane tunnels will not form between the branch sites. However, as shown in Figure 7-4, the OMP routes for all the prefixes that exist at all the branch sites will still be reflected to all of the other branch sites. Without having received the necessary TLOC advertisements, these OMP routes will be unresolvable and unused, but bandwidth and compute resources will be wasted in transmitting these network updates.

Technet24

Rather than continue to transmit routes that cannot be used, a better practice would be to filter out the routes from the branch sites in addition to the TLOCs. In order to accomplish this, select the + Sequence Type button and choose Route, as shown in Figure 7-12.

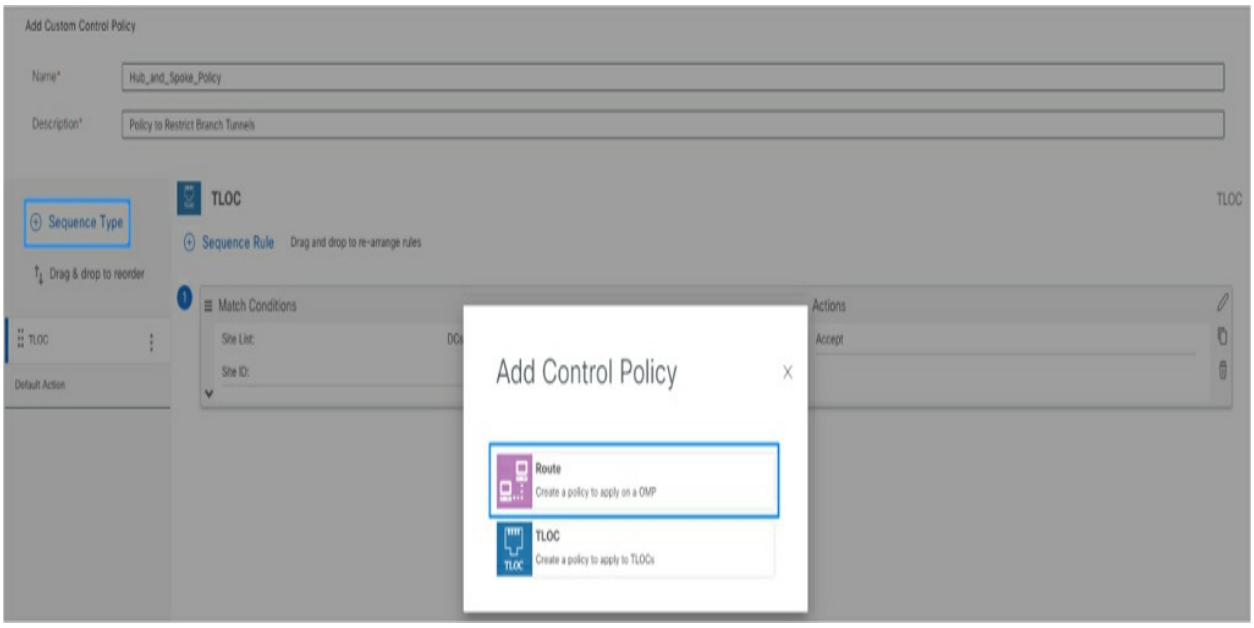


Figure 7-12 *Adding a Route Sequence to a Centralized Control Policy*

In a similar manner as was done for the TLOCs, the route sequence needs to be configured to match on the routes from the DCs site list and to accept them. No other sequences need to be configured at this time. You can see this configuration in Figure 7-13.

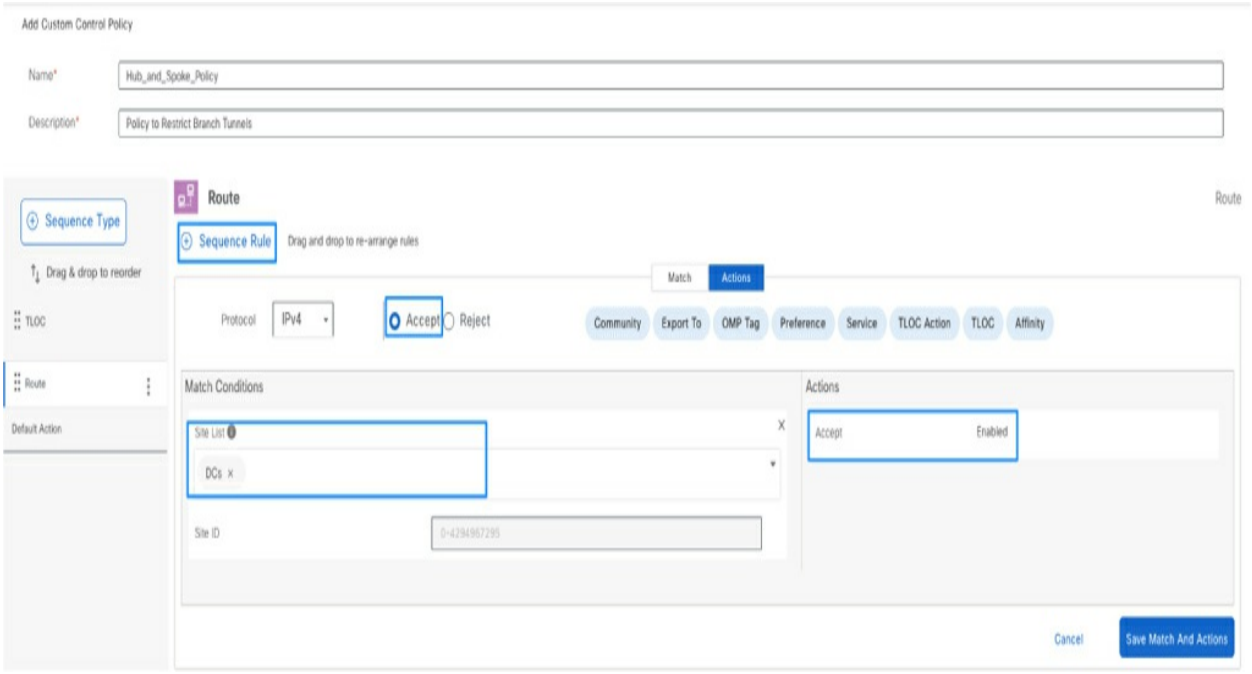


Figure 7-13 *Creating a Route Sequence Rule to Accept the DC Routes*

After the route sequence rule is saved by clicking the Save Match and Actions button, the entire policy can be saved by clicking the Save Control Policy button at the bottom.

Note

Sequence types are a subset of sequences. As the policy engine converts the visual policy to command syntax, the *topmost* sequence is processed, with its associated sequence rules. The CLI sequences start with the number 1 and increment by 10. Once that sequence is complete, it moves to the next sequence and its associated rule sets and so forth until the default policy action is processed. You can rearrange the order of the sequence types and sequence rules by dragging

Technet24

and dropping them into the desired processing order.

Once the control policy is saved, you can progress through the SD-WAN Policy Wizard by clicking Next through the Configure Topology and VPN Membership and Configure Traffic Rules screens, until you arrive at the Apply Policies to Sites and VPNs screen, shown in Figure 7-14. On this last page of the wizard, you need to configure a name for the policy that consists of a maximum of 127 alphanumeric characters, a hyphen or an underscore, and a description. In addition, on the Topology tab, you must specify where and in what direction the recently created Hub_and_Spoke_Policy should be applied. In this example, the policy will be applied in the SD-WAN Controller's outbound direction to the branch offices.

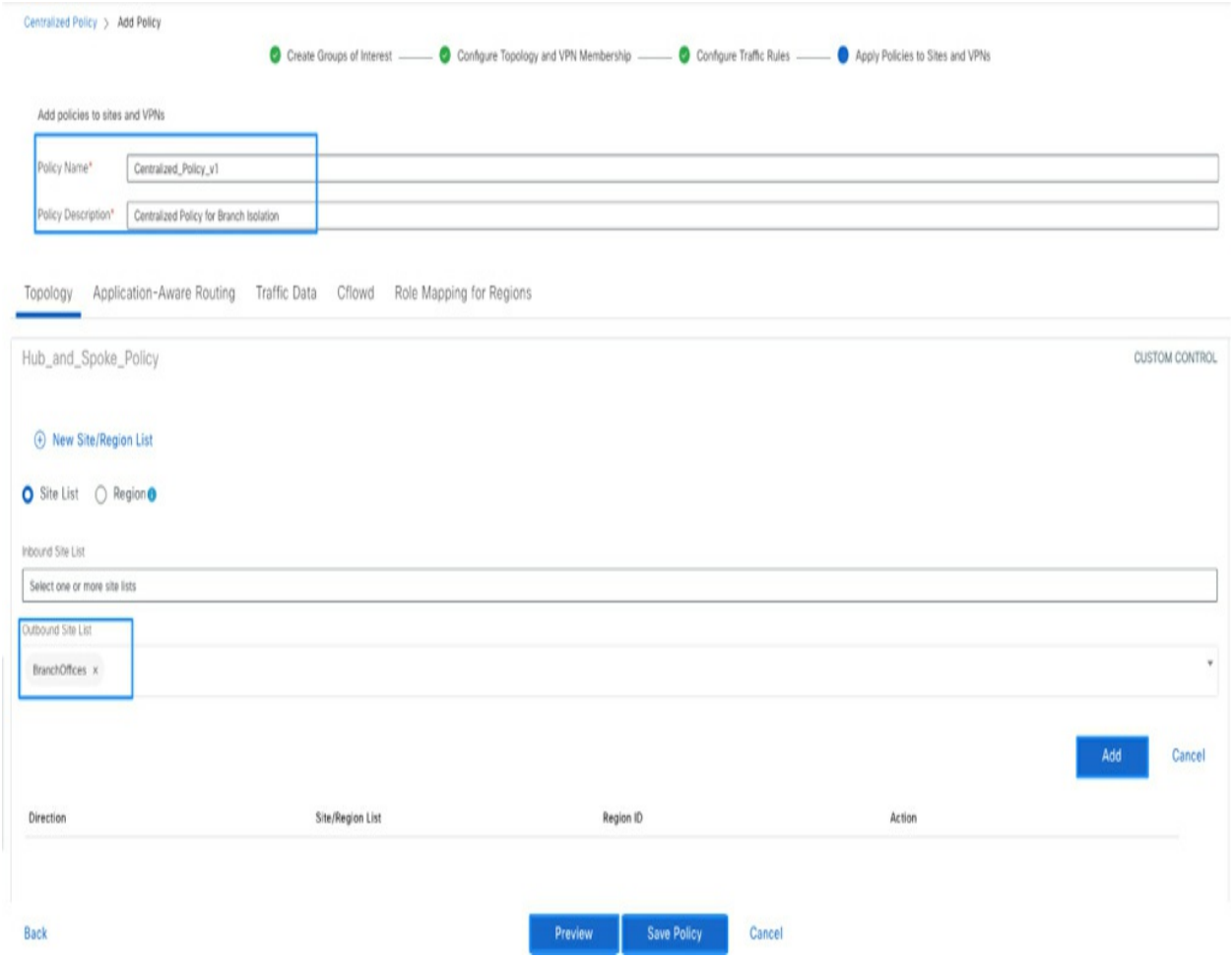


Figure 7-14 *Adding Policies to Sites and VPNs*

Only one global centralized policy can be applied at a time. This policy can contain different component policies, such as Topology, VPN membership, Application-Aware Routing (AAR), Data policy, and cFlowd. The last page of the wizard is used to link these component policies to the sites and VPN, where they will influence behavior.

Note

We recommend simply naming the centralized policies with

Technet24

intuitive names and version numbers, remembering to leave enough room to grow. In this case, the name Centralized_Policy_v1 was used as the policy name. As discussed in Chapter 6, there will only ever be a single centralized policy that is active at any point in time. From an operational view, it is easy to copy the existing centralized policy, increment the policy number, and then use the Policy Description field to list the changes from the previous version of the policy. In this way, you can create an archive of the policies on SD-WAN Manager. This way, should there ever be a need to roll back a change in the network, the process is as simple as activating a previous version of the policy.

Once the site list that the policy is to be applied to is saved by clicking the Add button, you can see the CLI of the entire policy by clicking the Preview button. Example 7-2 shows the full output of this policy.

Example 7-2 *Use Case 1: Complete Centralized Policy to Create Hub-and-Spoke Topology*

```
policy
 control-policy Hub_and_Spoke_Policy
    sequence 1
```

```
      match tloc
       site-list DCs
      !
      action accept
      !
     !
! The reference to the prefix list below was nc
! but instead was added automatically by SD-WAN
! logic remains the same: Match all Routes.
    sequence 11
     match route
      site-list DCs
      prefix-list _AnyIpv4PrefixList
     !
     action accept
     !
    !
  default-action reject
 !
 lists
  site-list BranchOffices
   site-id 100-199
  !
! Only Routes and TLOCs that match this site li
! by the policy above.
```

Technet24

```
    site-list DCs
     site-id 10-50
    !
    prefix-list _AnyIpv4PrefixList
     ip-prefix 0.0.0.0/0 le 32
    !
   !
  !
  ! The policy is applied in the outbound directi
  ! match the site list "BranchOffices".
  apply-policy
   site-list BranchOffices
    control-policy Hub_and_Spoke_Policy out
   !
  !
```

The policy can then be saved by clicking the Save Policy button at the bottom of the page with the configuration preview. Once the policy is saved, it can be applied to the SD-WAN fabric by selecting the Activate option from the policy menu, as shown in Figure 7-15. As discussed in Chapter 6, in the process of activating a policy, SD-WAN Manager writes the policy into the configuration of the SD-WAN Controllers.

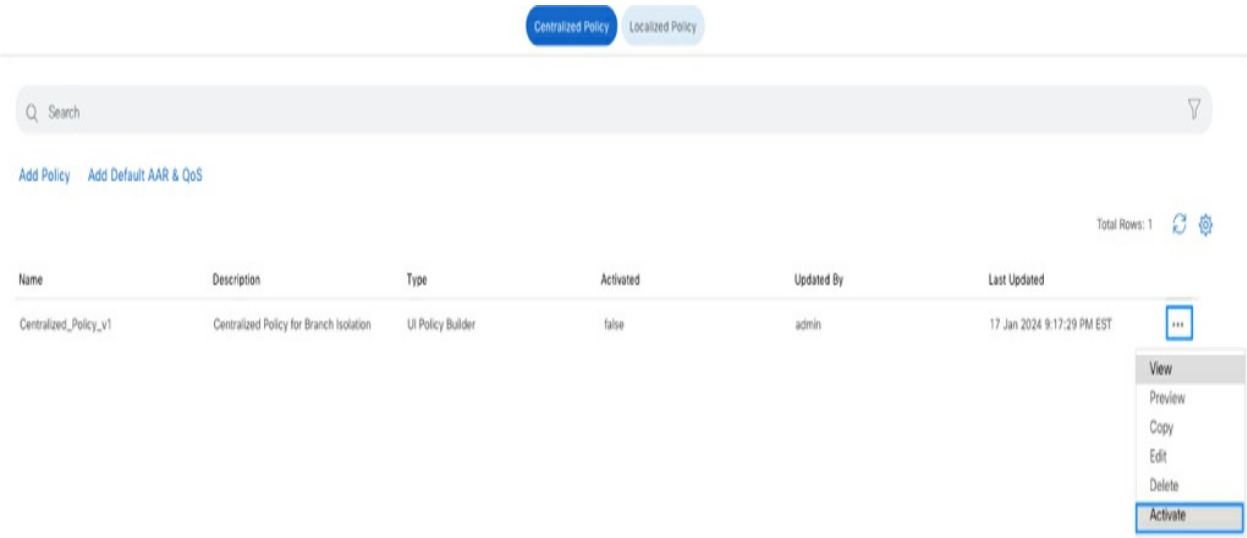


Figure 7-15 *Activating a Centralized Policy*

Note

Because configuring policy on a SD-WAN Controller from SD-WAN Manager is actually manipulating the configuration of the SD-WAN Controller itself, the SD-WAN Controllers must be in SD-WAN Manager (vManage) mode and have a template applied from SD-WAN Manager. This allows SD-WAN Manager to have authoritative control of the SD-WAN Controller configuration.

Topology Workflow Approach

The topology methodology of influencing a behavior at the SD-WAN Controller level is available starting with SD-WAN Release 20.12. It allows you to provision mesh or hub-and-spoke topologies with modular, reusable

Technet24

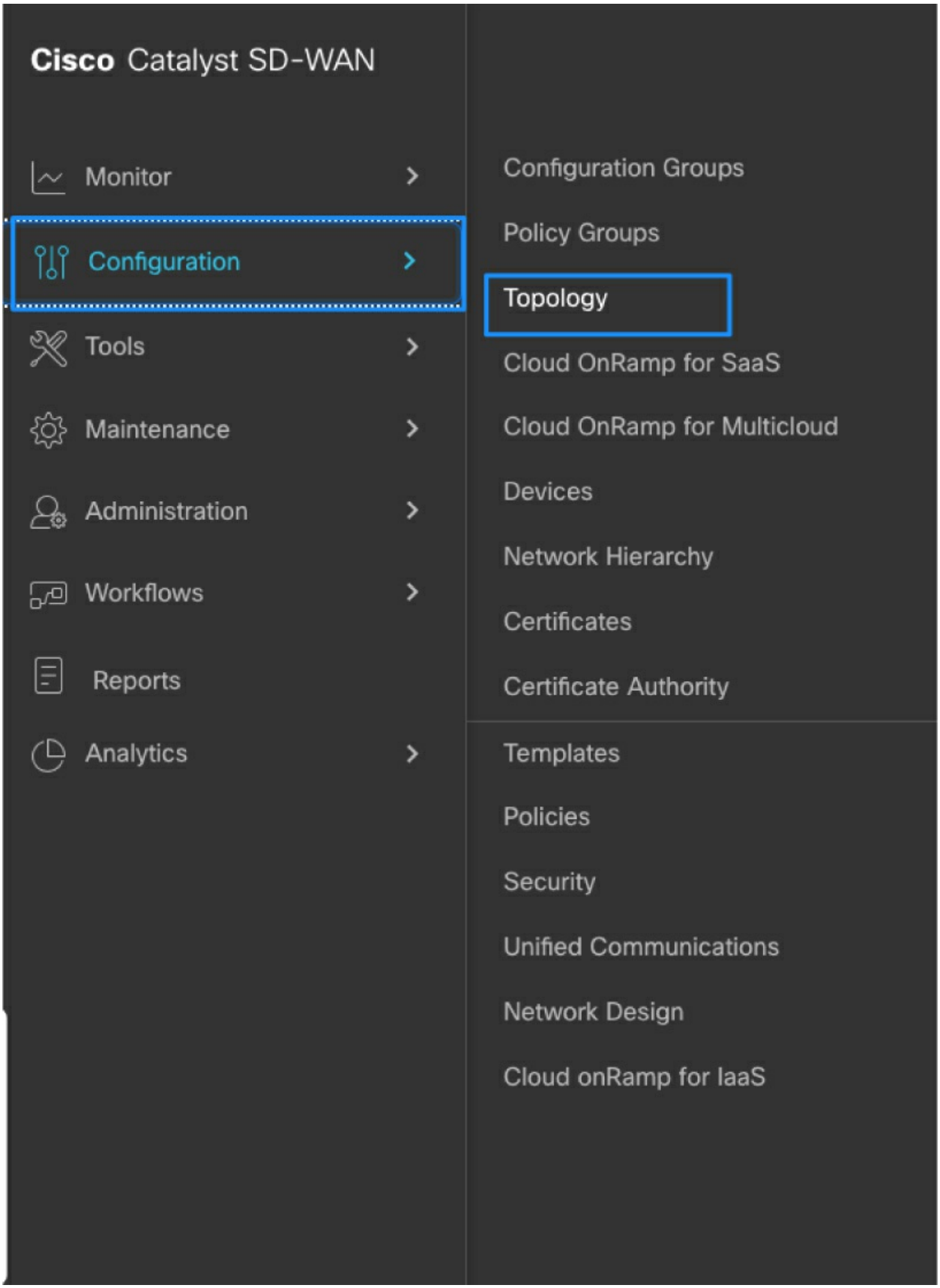
logic that is applied at one or more sites; this logic can be added, edited, or removed easily based on site location, in contrast with the traditional approach that governs all sites. (You can find the prerequisites for this approach in the official documentation “Policy Groups Configuration Guide.”)

You can leverage this policy through a predefined topologies library that includes mesh and hub-and-spoke topologies, or you can create a custom topology, as in the traditional control policies case.

Note

Starting from SD-WAN Releases 20.12 and 17.12, the hub-and-spoke configuration approach also streamlines the setup of this type of topology, but it eliminates the need for a centralized control policy. Instead, it requires a simple configuration command at each SD-WAN Controller, hub router, and spoke router. For detailed information, refer to “Cisco Catalyst SD-WAN Routing Configuration Guide.”

The initial step in constructing this policy through the first option is to open the Topology Wizard by clicking Configuration > Topology in SD-WAN Manager, as shown in Figure 7-16. (Notice that the following figures present both the guided workflow solution and a custom topology.)


Cisco Catalyst SD-WAN
Monitor
Configuration
Tools
Maintenance
Administration
Workflows
Reports
Analytics
Configuration Groups
Policy Groups
Topology
Cloud OnRamp for SaaS
Cloud OnRamp for Multicloud
Devices
Network Hierarchy
Certificates
Certificate Authority
Templates
Policies
Security
Unified Communications
Network Design
Cloud onRamp for IaaS

Technet24

Figure 7-16 *Getting into the Topology Wizard*

In the Topology Wizard, you need to select Create Topology, as show in Figure 7-17, and provide a name and description for the topology policy, as shown in Figure 7-18. Then click Create.



Figure 7-17 *Creating a Topology*



Figure 7-18 *Providing a Name and Description for the Topology*

After creating the topology, click Add Topology to reveal a drop-down menu, where Hub and Spoke is selected, to start creating the logic to isolate the communication between branches (see Figure 7-19).

Technet24

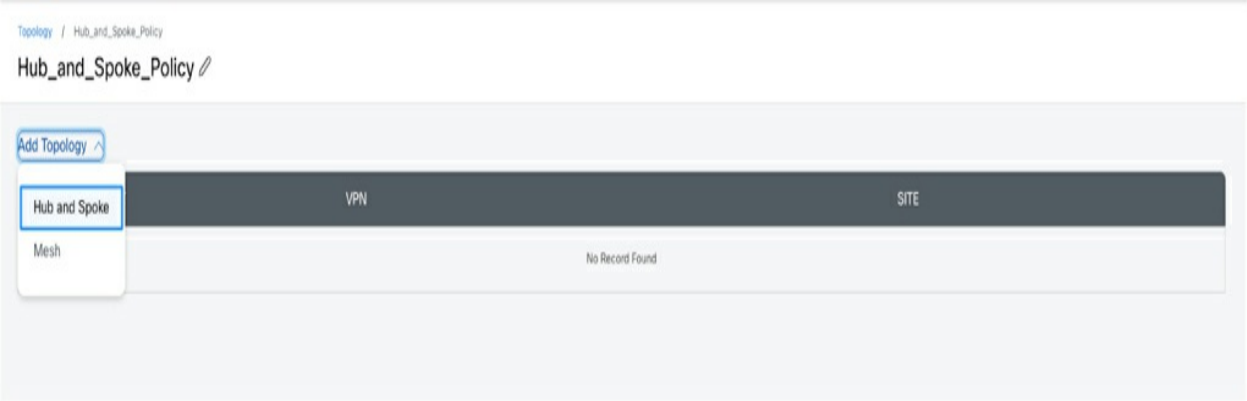


Figure 7-19 *Hub and Spoke Guided Workflow Selection*

Figure 7-20 shows that your next step is to provide a name for the policy (in this case, Hub_and_Spoke_Policy) and select the data segment (VPN) that requires the topology enforcement.

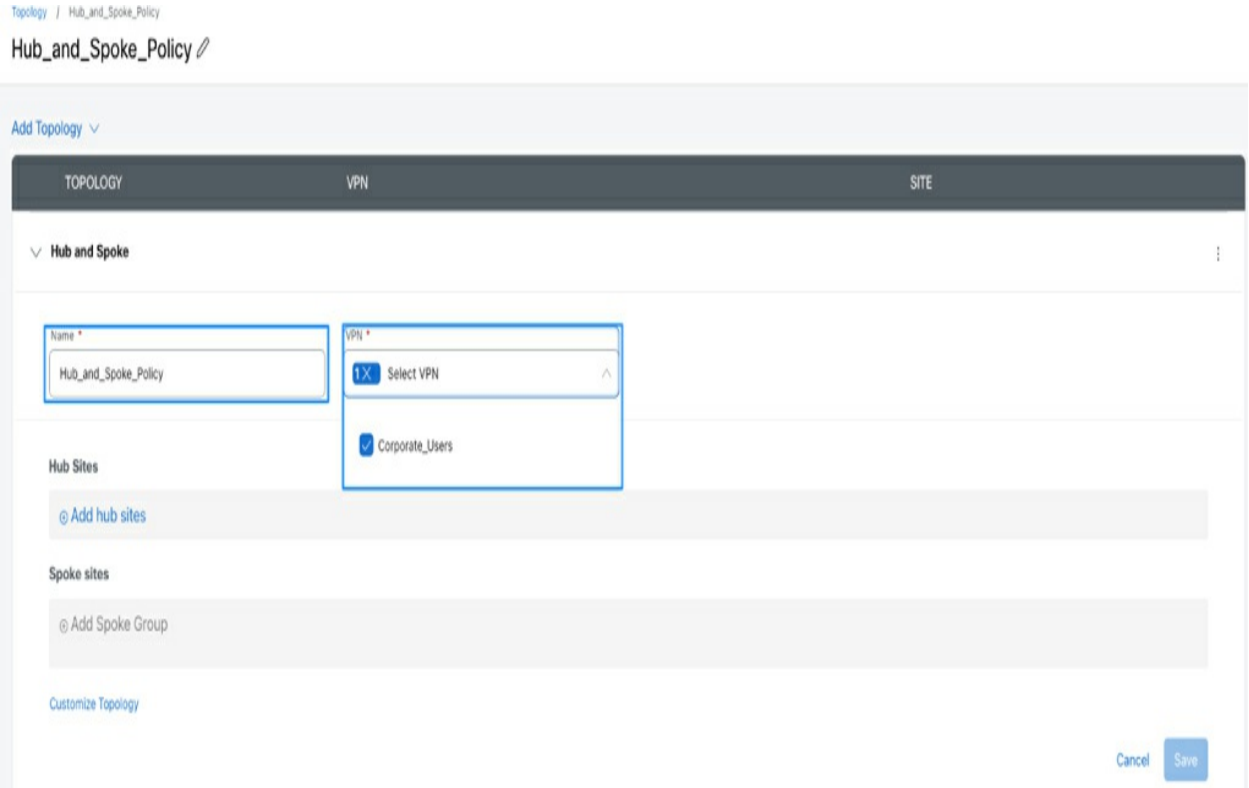


Figure 7-20 *Adding a Name and Selecting the Data Segment*

Select the appropriate site ID boxes of the sites to be used as Hub and Spokes respectively in the lists shown in Figure 7-21 and Figure 7-22.

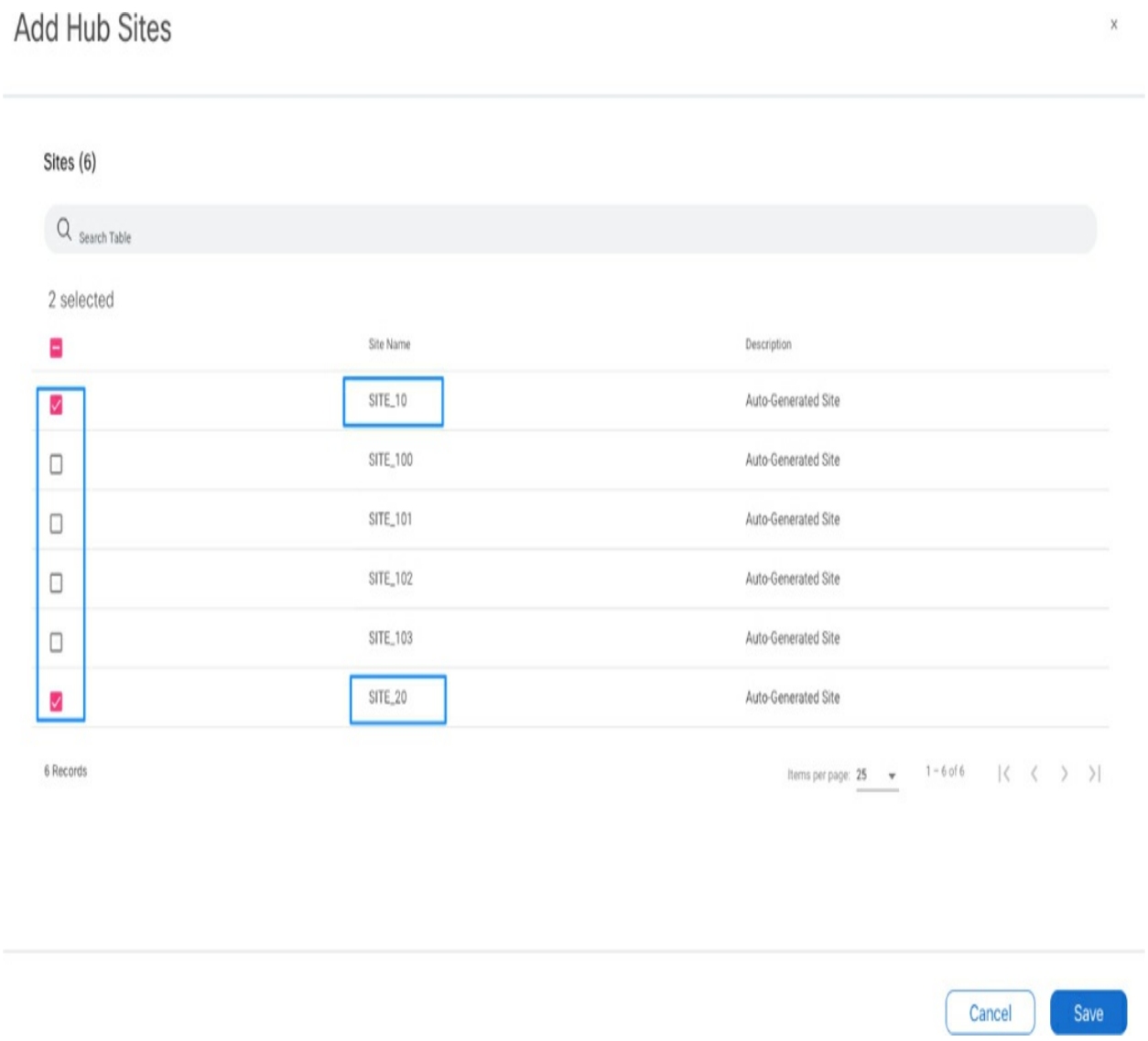


Figure 7-21 *Checking Site Boxes to Be Used as Hubs*

Technet24

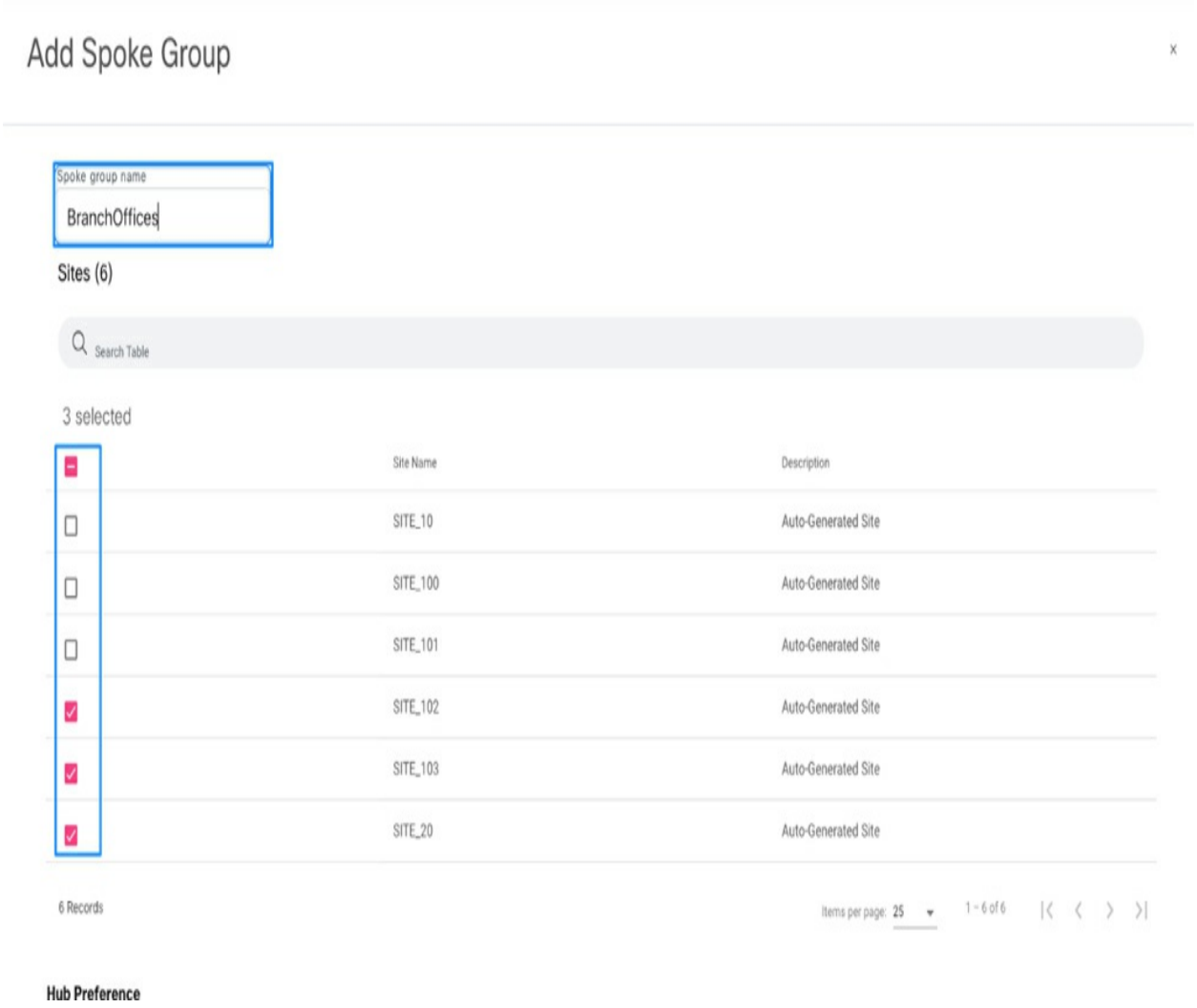


Figure 7-22 *Checking Site Boxes to Be Used as Spokes*

Note

Both data segments (VPNs) and site names available in this step are automatically created when the Configuration Groups workflow is used. Under Configuration > Network Hierarchy, you can manage SD-WAN fabric sites information such as name, description, and location parameters to simplify the overall experience. For more information, refer to the “System

and Interfaces Configuration Guide.”

In the Spoke Group selection screen, you can set a Hub Preference to specify the priority (see Figure 7-23). In this case, the data center sites are assigned the same priority, which means traffic will be load balanced. Click Save.

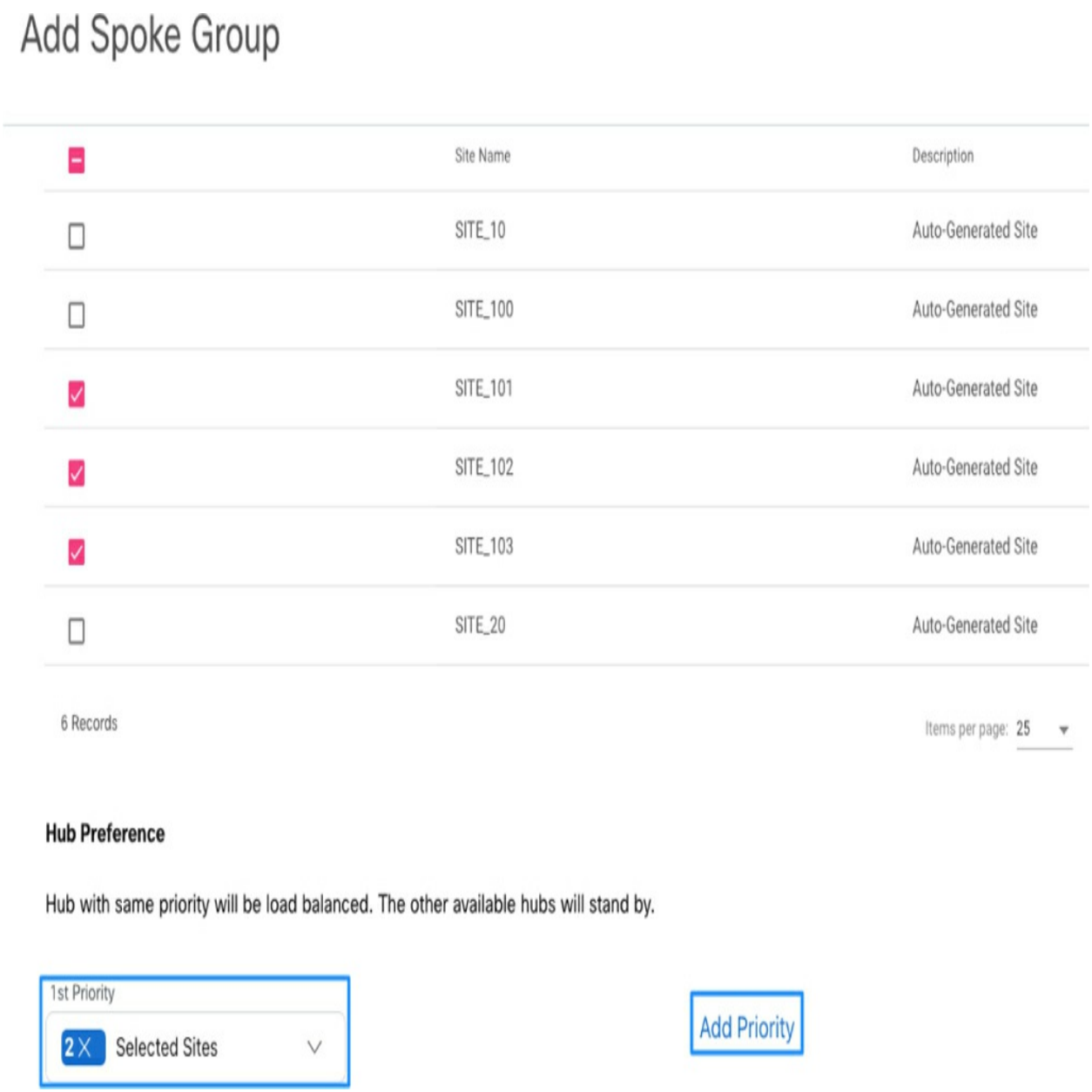


Figure 7-23 *Hub Sites Priority Setting*

Technet24

As you can see in Figure 7-24, a summary screen shows the values previously populated through the guide workflow for the hub-and-spoke topology. Click Save to complete the configuration.

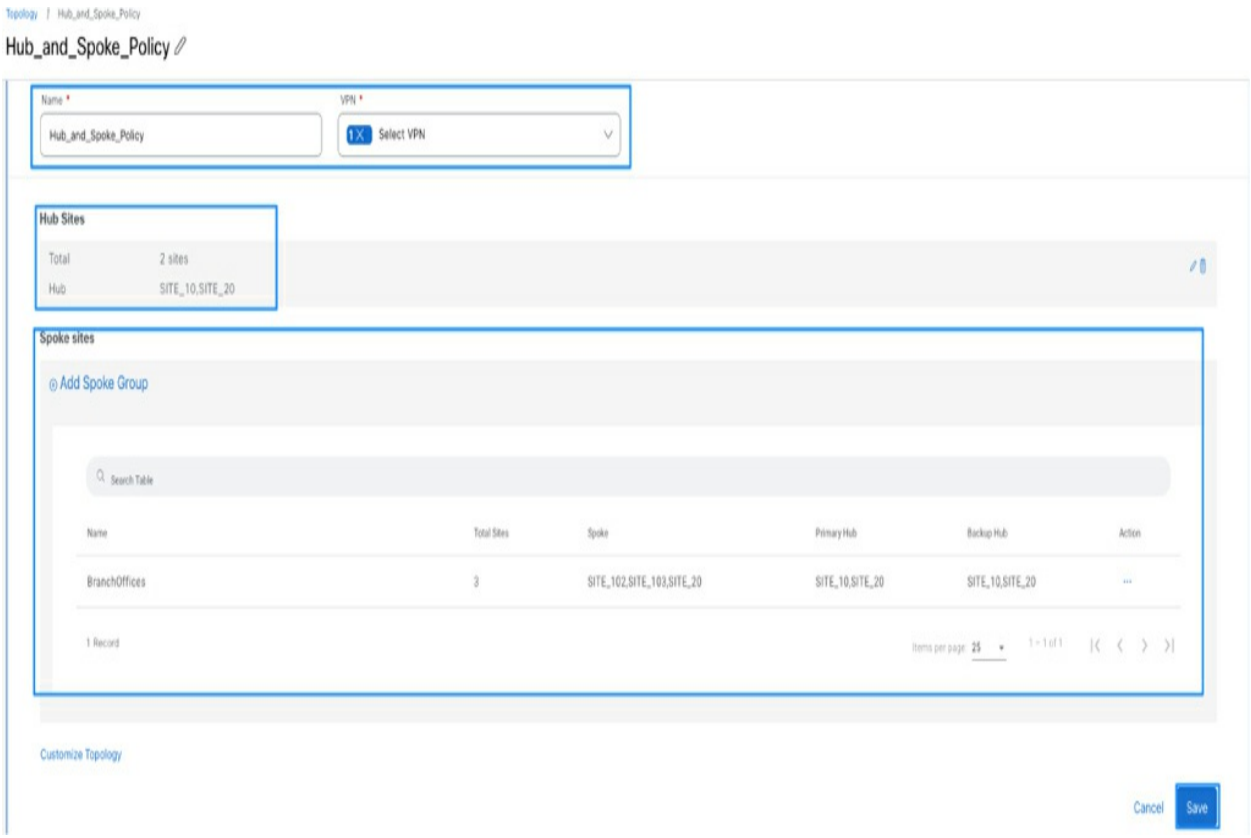


Figure 7-24 *Summary Screen for the Topology Group*

The previous workflow result is ready to influence the routing behavior when activated. This helps an organization quickly achieve the goal of applying a particular topology to the fabric, but you can see that there are not too many customizations available. Figure 7-25 shows the policy activation procedure. Remember that an SD-WAN Controller needs to be attached to a template in order for SD-WAN Manager to have authoritative control of the configuration.

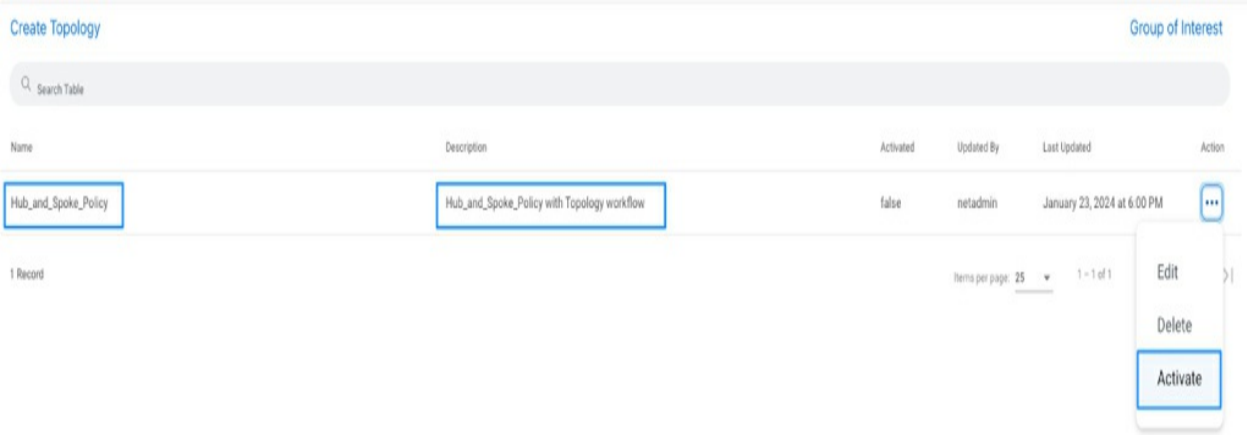


Figure 7-25 *Topology Activation*

When granular match-action pairings are required, using a custom topology is the best approach. You can start a topology from scratch or customize an existing one by selecting Customize Topology, as shown in Figure 7-26.

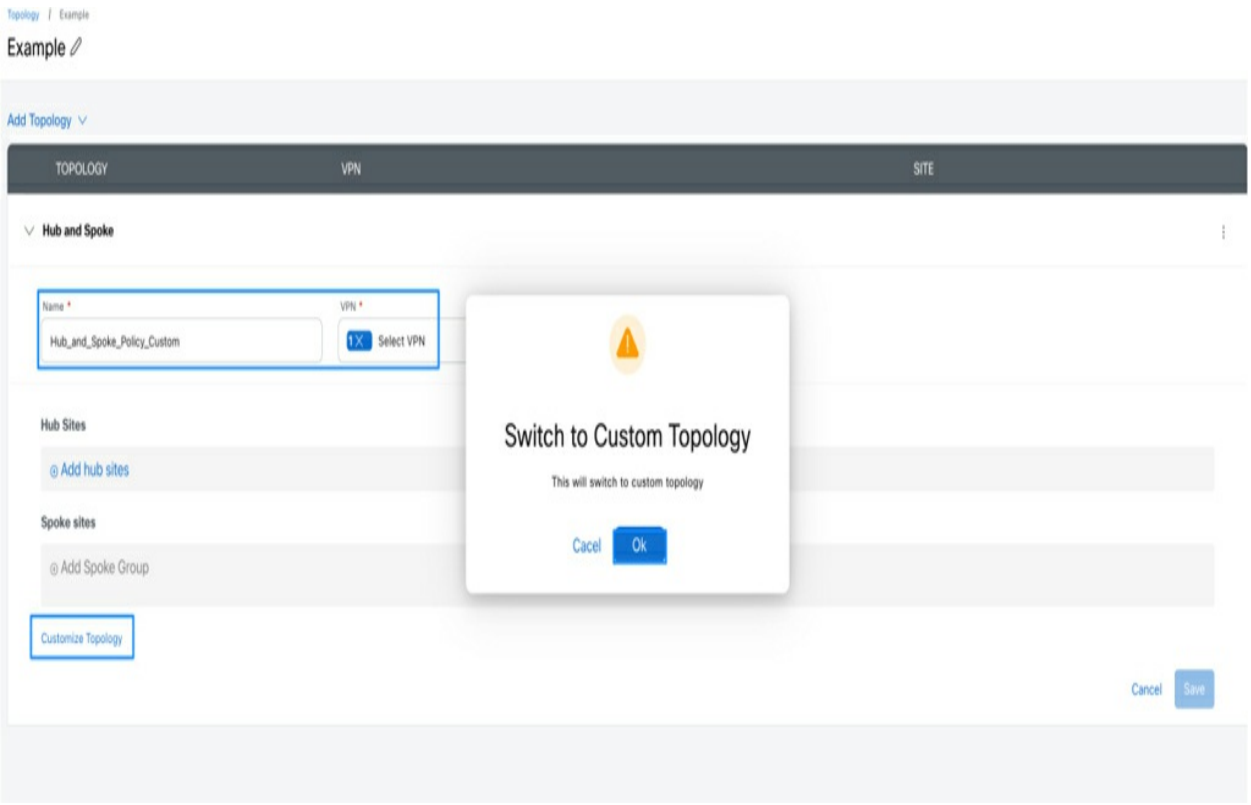


Figure 7-26 *Switching to a Custom Topology*

Technet24

As shown in Figure 7-27, the first two parameters to provide for the custom topology are the name and VPN. As shown in Figure 7-28, the third parameter to specify is for Outbound Sites (which in traditional control policies means the outbound direction).

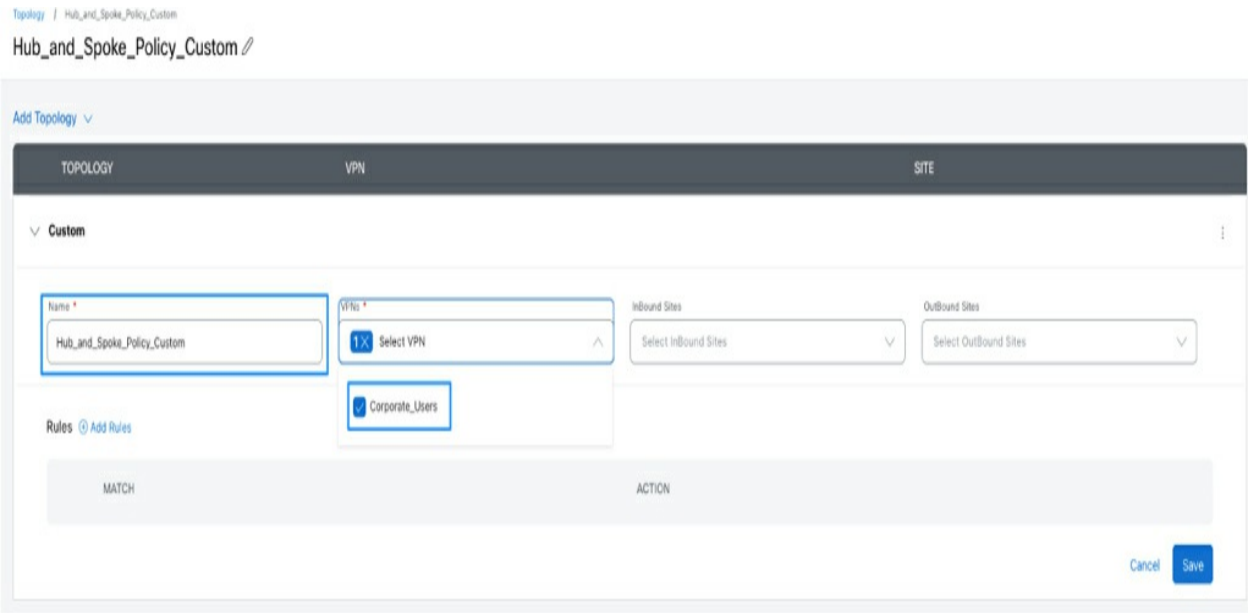


Figure 7-27 *Selecting the Name and Data Segment*

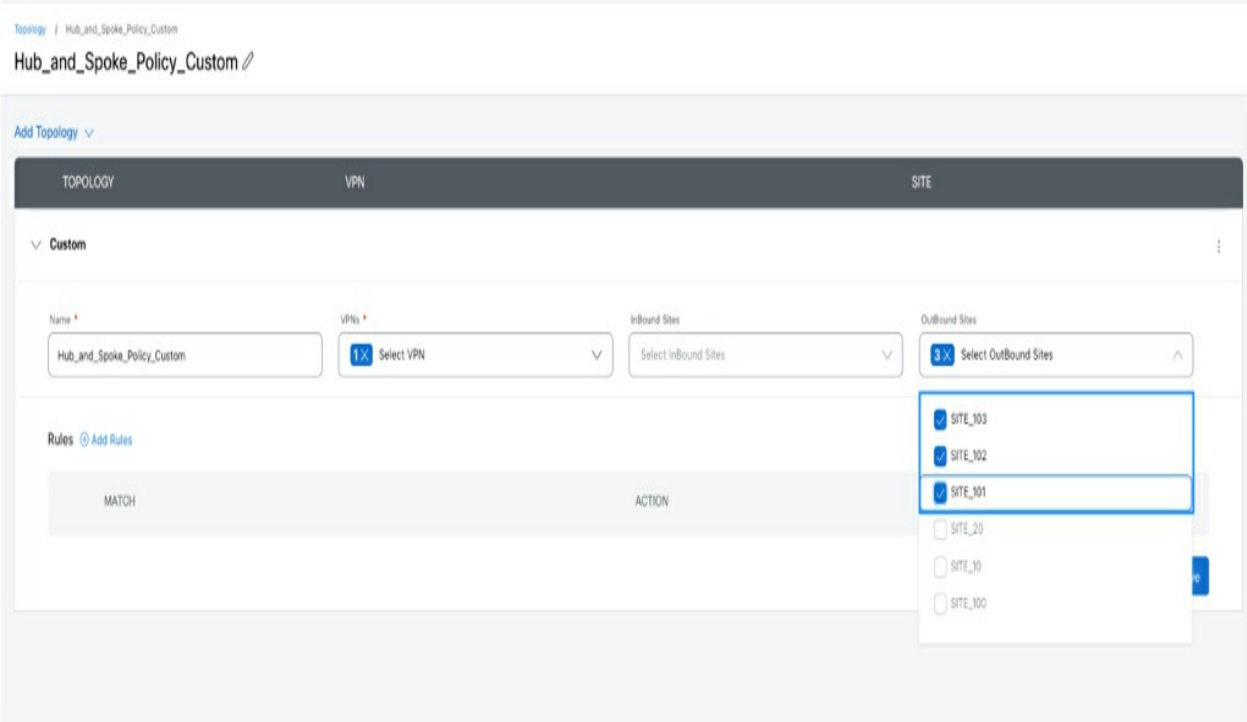


Figure 7-28 *Selecting Outbound Sites*

At this point, the policy direction and application are set, but the logic still needs to be set. Use the following steps to add match-action rules (see Figure 7-29 and Figure 7-30):

Click **Add Rules**. Sequence 1 is automatically populated.

Provide a name—in this case, **Match_and_Accept_DC_TLOCS**. Set the type to **TLOC**.

Select **Add Conditions** and check the boxes for the DC1 and DC2 WAN Edge devices (site IDs 10 and 20).

Change Action to **Accept** and click **Save Match and Actions**.

Technet24

Click **Add Rules**. Sequence 2 is automatically populated.

Provide a name—in this case, **Match_and_Accept_DC_Routes**. Set the type to **Route**.

Select **Add Conditions** and check the boxes for the DC1 and DC2 WAN Edge devices (site IDs 10 and 20).

Change Action to Accept and click Save Match and Actions.

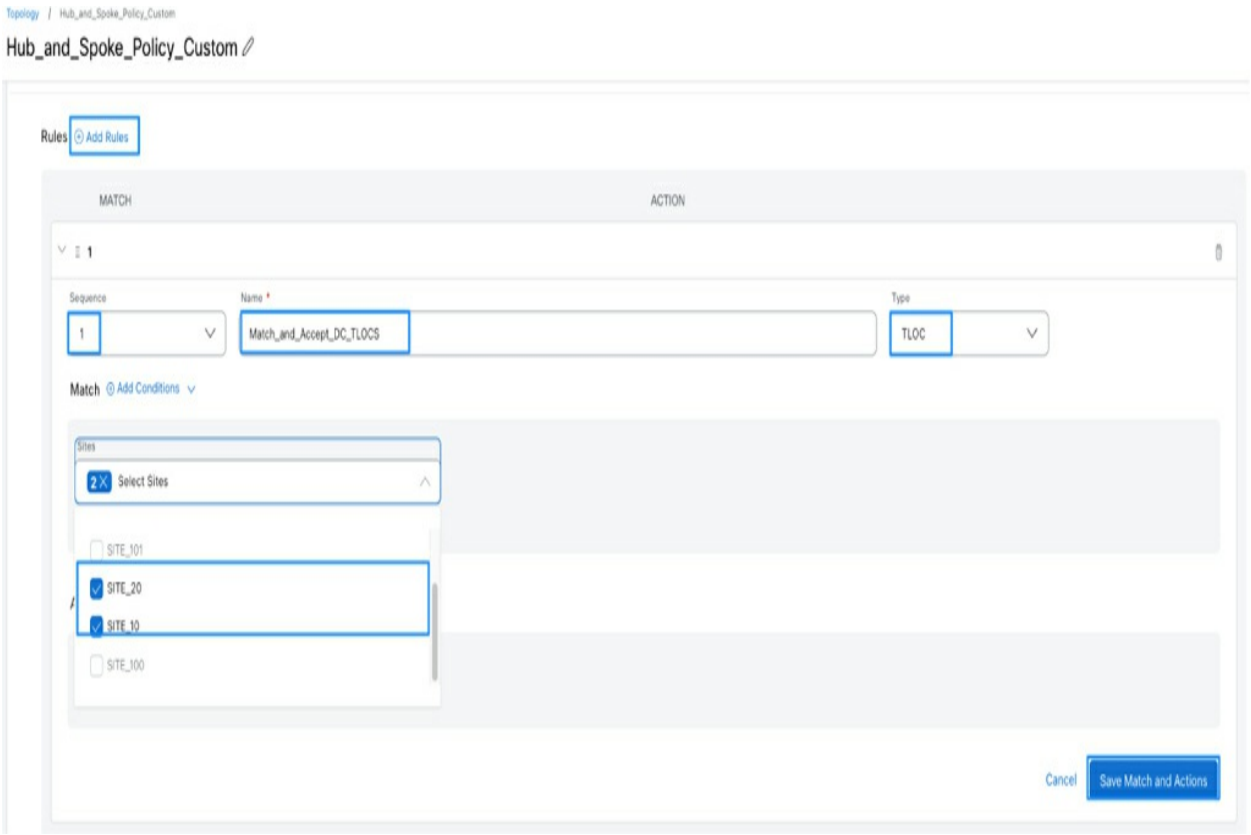


Figure 7-29 *TLOC Type Match-Action Configuration*

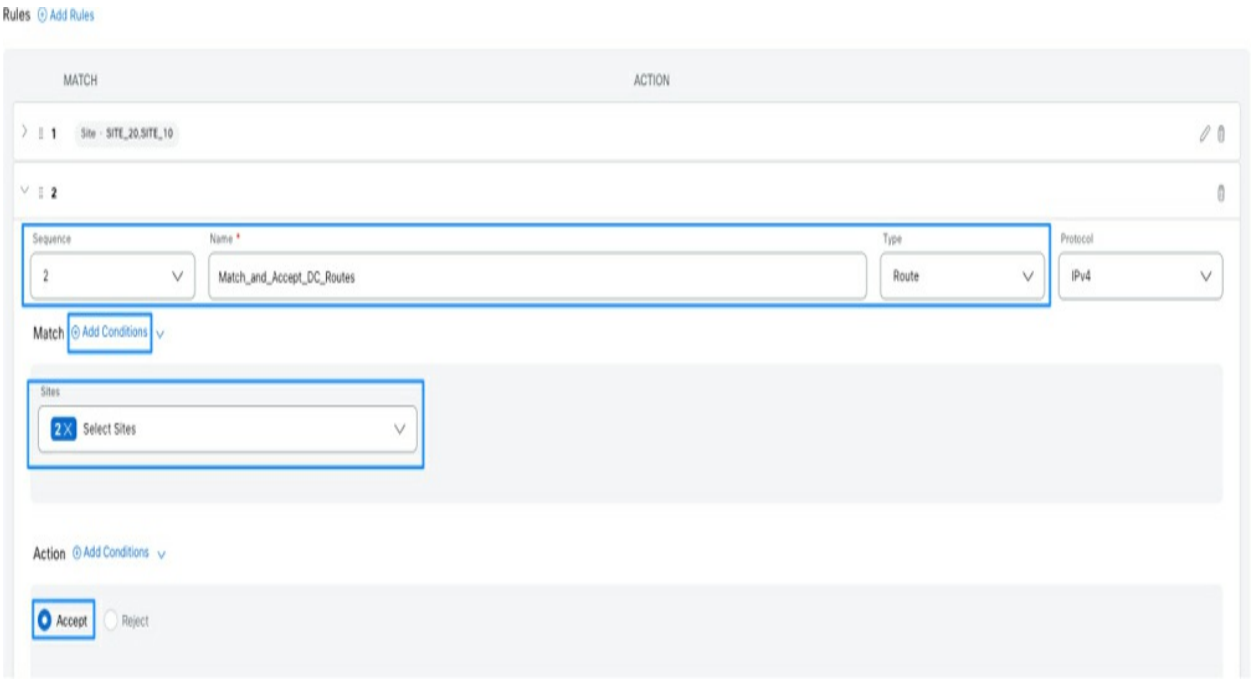


Figure 7-30 *Route Type Match-Action Configuration*

To complete the custom policy setup, click Save (see Figure 7-31). This screen provides a summary of the rules previously configured for the custom topology.

Technet24

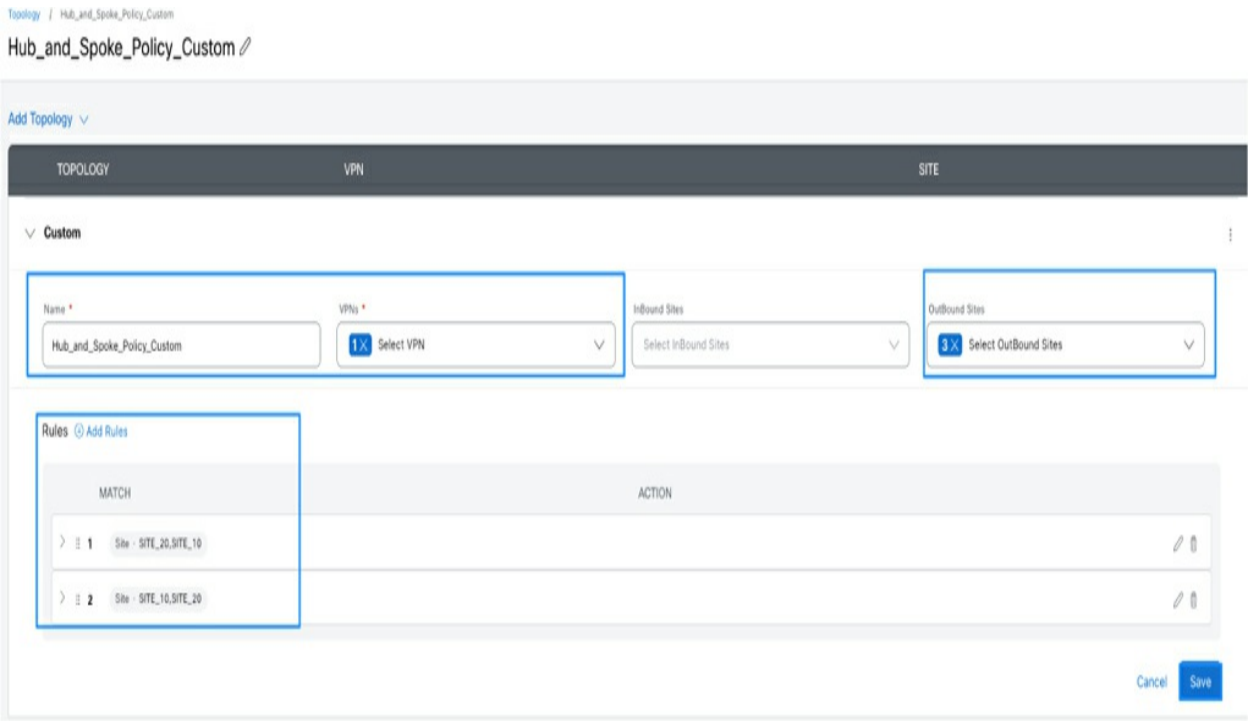


Figure 7-31 *Custom Topology Summary*

To activate a policy, select Activate next to the policy, as shown in Figure 7-32.

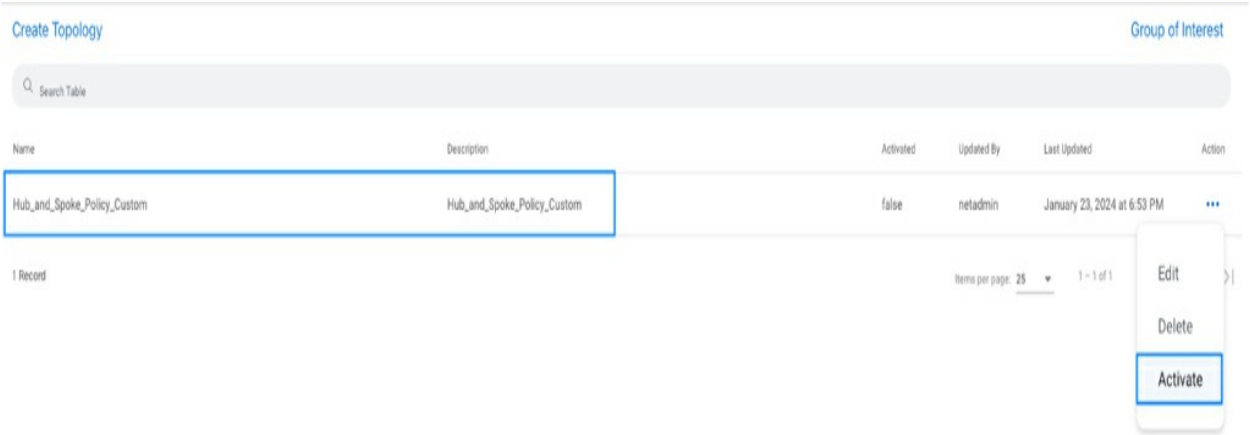


Figure 7-32 *Policy Activation*

Once the policy has been applied, you can see the effects of the policy by using the same monitoring techniques used before the policies were applied.

From Monitor > Devices > BR2-Edge1 > Real Time, you can see the current list of BFD sessions (refer to Figure 7-16). When you compare Figure 7-33 to Figure 7-3, which shows the output before the policy was applied, you can see that the policy resulted in the number of BFD sessions (and IPsec tunnels) decreasing from 24 to 12. If you look closely at Figure 7-33, you also see that all the remaining BFD sessions come from routers with system IP addresses 10.0.10.1, 10.0.10.2, 10.0.20.1, or 10.0.20.2 (for routers in Data Center 1 or Data Center 2). All these tunnels are also established to sites with either Site ID 10 or 20.

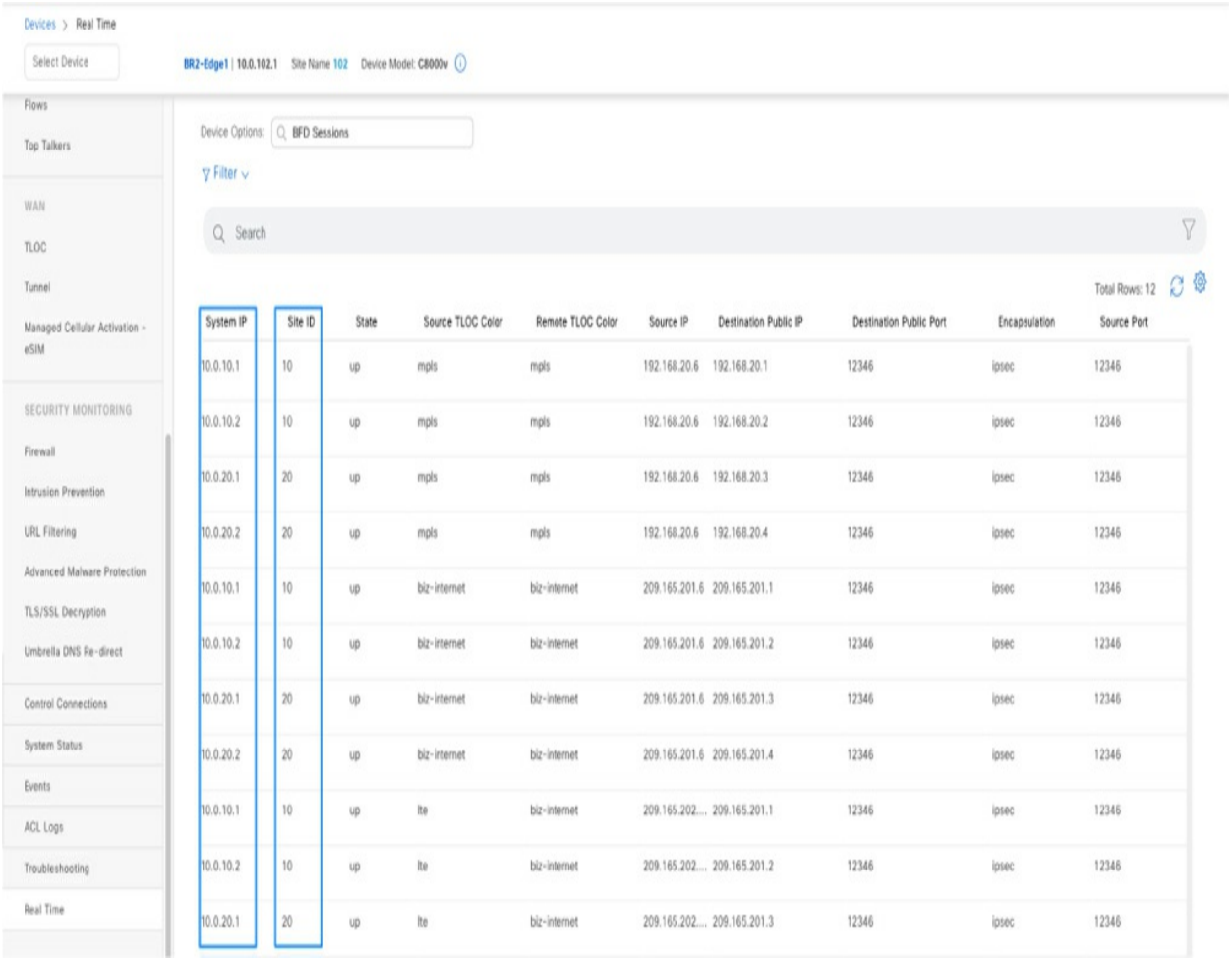


Figure 7-33 *BR2-Edge1 BFD Sessions After the Hub-and-Spoke Policy Is*

Technet24

Applied

Before the policy was applied, you were able to run a **traceroute** from Branch 2 to Branch 3 directly (refer to Example 7-1). Now, when the same **traceroute** is run from BR2-Edge1, you can see that the network is unreachable, as signified by the "* * *" response from the local host in Example 7-3.

Example 7-3 *The Network Is Unreachable When Tracing from BR2-Edge1 to BR3-Edge1*

```
! Traceroute from BR2-Edge1 to BR3-Edge1:
! The traceroute is unsuccessful based on the
BR2-Edge1# traceroute vrf 10 10.10.103.1
<<<Output omitted for brevity>>>
  1  *  *  *
  2  *  *  *
```

```
! Traceroute from BR2-Edge1 to DC1-Edge1:
! The traceroute is successful; resources in th
BR2-Edge1# traceroute vrf 10 10.10.10.100
<<<Output omitted for brevity>>>
```

```
  1 MPLS-DC1-Edge1 (192.168.20.1) 2 msec 1 msec
  2 DC1-Edge1-Core (172.16.10.2) 1 msec 2 msec
  3 DC1-Service (10.10.10.100) 2 msec
BR2-Edge1#
```

The second **traceroute** to the data center remains successful after the policy is applied. This behavior can be further explained by examining the IP routes table in the real-time output of BR2-Edge1. As shown in Figure 7-34, after filtering for routes only in VPN 10, you can see that the branch office no longer has a route for the 10.10.103.0/24 prefix, and therefore the network is listed as unreachable. The only routes that remain in the VPN 10 routing table on BR2-Edge1—other than the locally connected route 10.10.102.0/24—are the routes with prefixes 10.10.10.X and 10.10.20.X that originate in the data centers (Site ID 10 and Site ID 20).

Technet24

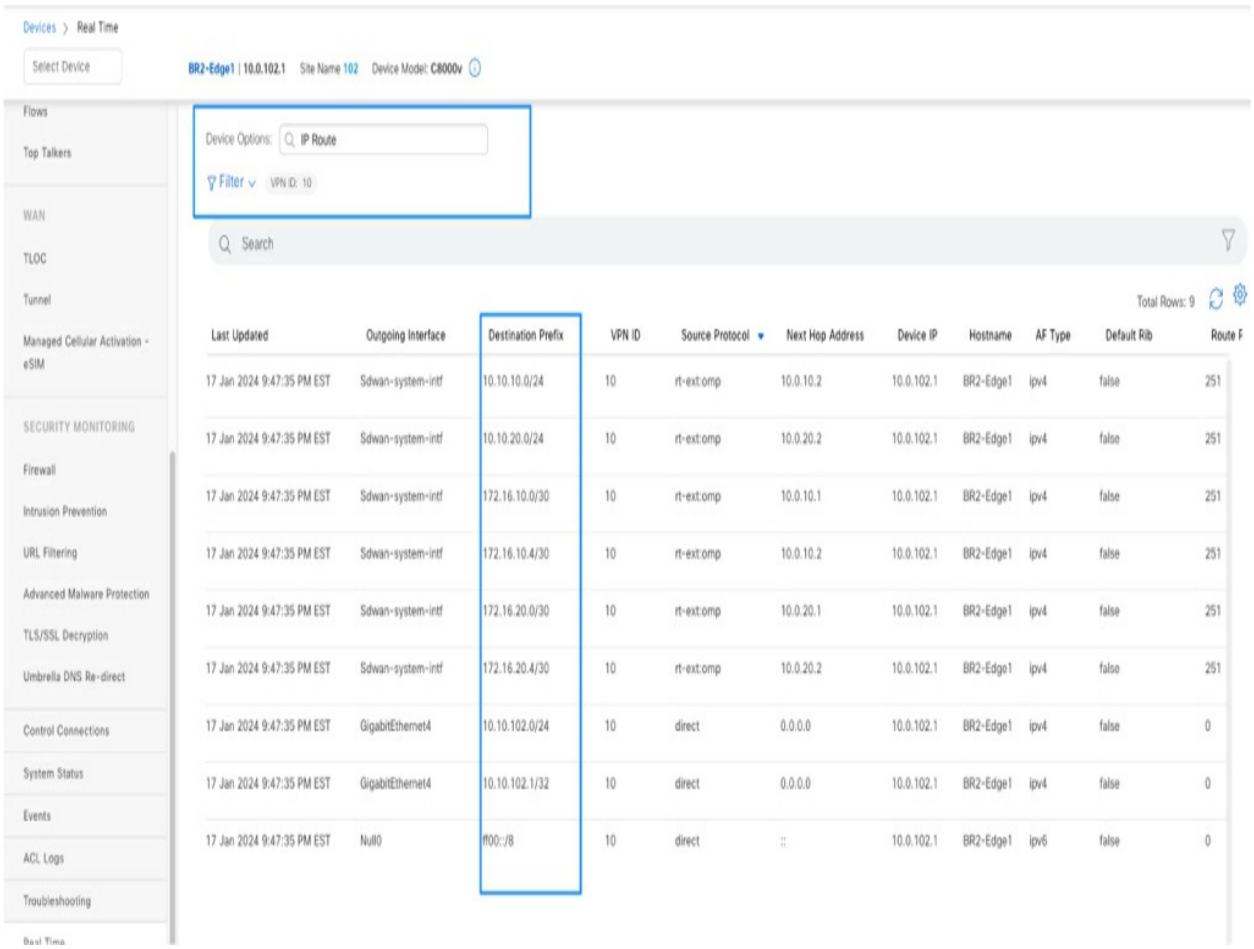


Figure 7-34 *The BR2-Edge1 Routing Table After the Policy Is Applied*

Use Case 1 Review

In this use case, you used centralized control policies to prohibit site-to-site communication between branch locations. By encoding the business intent into the structure of the SD-WAN fabric in this fashion, you can harden the security posture of the network by preventing unintended east–west traffic flows between branch sites. The use case explores the two existing methods to achieve this goal. We recommend selecting either traditional control policies contained in centralized policies or newer topology policies in order to avoid operational complexity.

Use Case 2: Enabling Branch-to-Branch Communication Through Data Centers

In the previous use case, you saw how centralized control policies can be used to completely isolate branch sites from each other. While this may be appropriate for some use cases where no communication is preferred, there are also use cases where organizations may want to enable site-to-site communication indirectly by proxying traffic through a data center or some other regional hub. This can have the benefits of reducing complexity and the scale of the control and data planes at the network edge while still maintaining full connectivity for the occasional traffic flows that may need to exist between branch locations. We will explore two different potential solutions for this use case: summarization and TLOC lists.

Enabling Branch-to-Branch Communication with Summarization

Summarization, in addition to the centralized control policy discussed in the previous use case, builds on the principle of longest-match routing and is not specific to any technology in Cisco SD-WAN. If a network summary (such as 10.0.0.0/8) or the default route (0.0.0.0/0) were advertised from the data center routers, traffic would follow the path to the data center WAN Edge routers. The data center routers would then examine their routing tables and follow the route to the more specific match advertised from the branch sites. This is a common design mechanism that is used in hub-and-spoke WAN deployments such as with DMVPN Phase 1. Figure 7-35 illustrates this

Technet24

process.

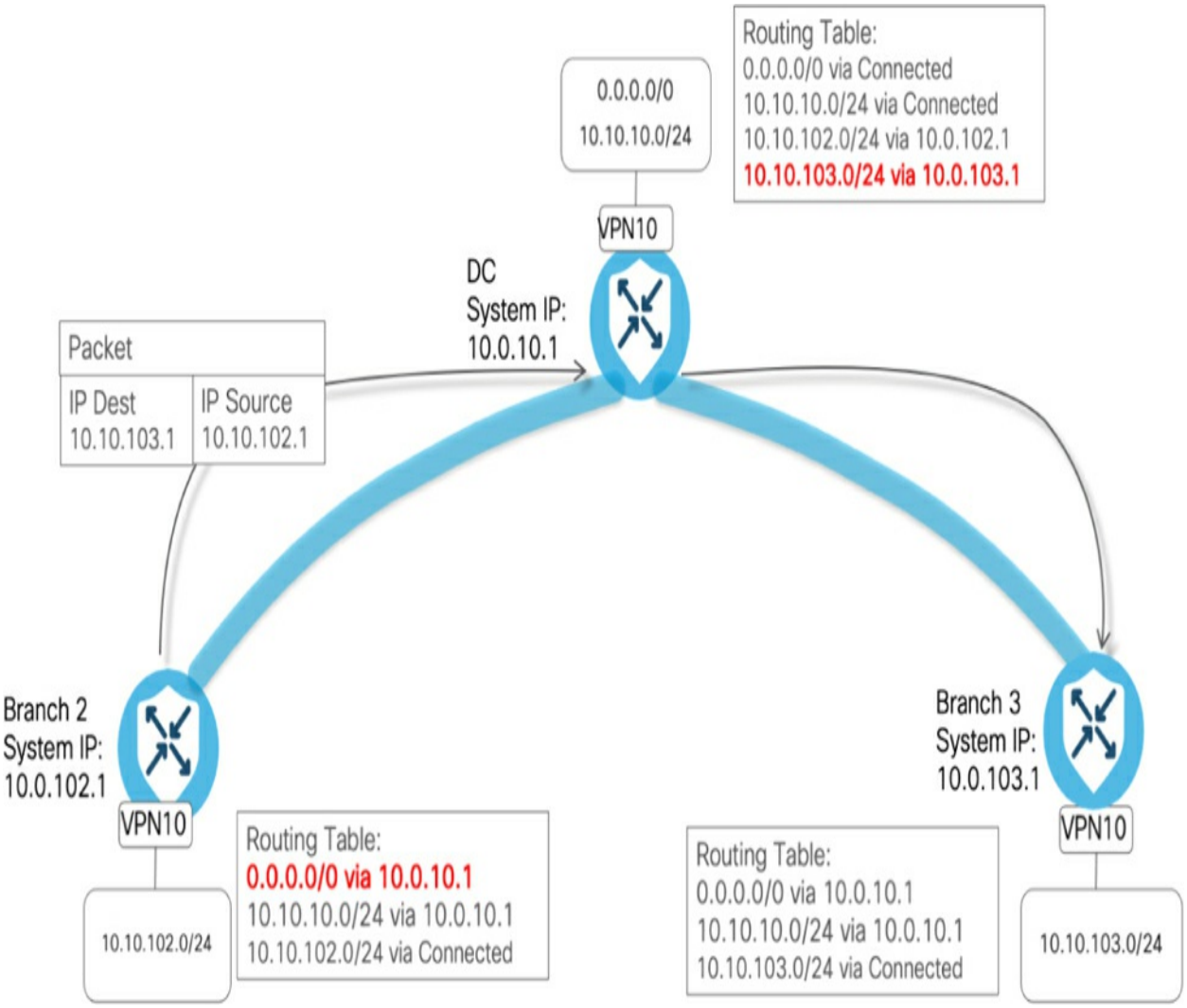


Figure 7-35 *Enabling Branch-to-Branch Communication with Summarization*

As shown in Figure 7-35, when the WAN Edge in Branch 2 does a routing lookup in order to forward a packet to the destination 10.10.103.1, the most specific match in the routing table is the default route. The default route is being advertised from the WAN Edge router in the data center with system IP address 10.0.10.1, so the packet is forwarded to that router. When the packet

arrives at the data center router, it performs a lookup on the destination 10.10.103.1 in its own routing table. As the data center router has not had any routes or TLOC routes filtered, it has a more specific route to 10.10.103.1 via 10.0.103.1 and forwards the packet along to Branch 3. So you can establish communication through the data centers by injecting a default or summary route, without needing to change the SD-WAN fabric or any policies.

For this use case, say that you have injected a default route in Data Center 1 that is being advertised by DC1-Edge1 and DC1-Edge2. The specific method used to inject the default route is unimportant. If you have a default route in the routing protocol running in the data center, it can be advertised directly into OMP. Alternatively, you can configure a static default route and advertise that into OMP. Regardless of the method used, once the default route is advertised into OMP, it is then propagated to the branch offices, as shown in Figure 7-36.

Technet24

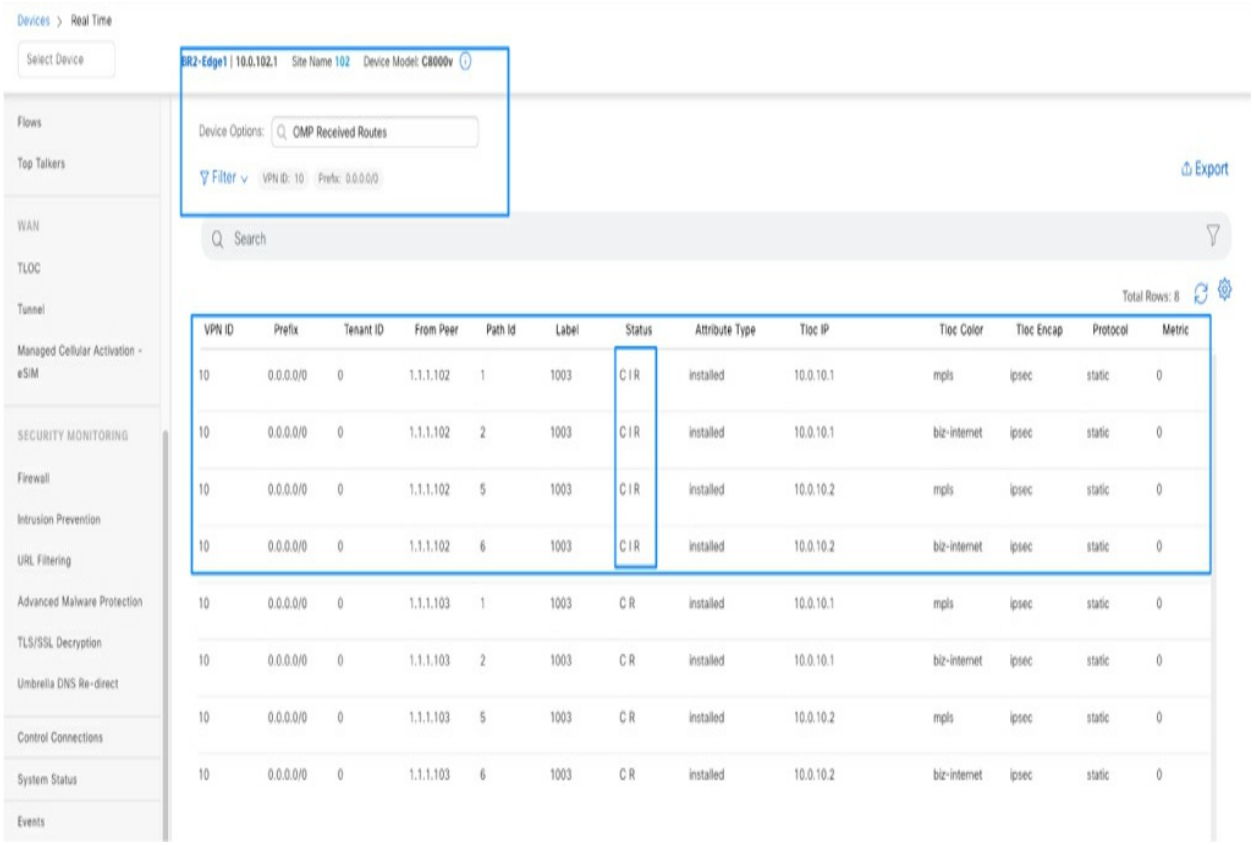


Figure 7-36 *BR2-Edge1 Routing Table with Default Route*

With the default routes in place, traffic can now be forwarded to the data center and then to a different branch site, as shown in Example 7-4.

Example 7-4 *Tracing from BR2-Edge1 to BR3-Edge1 Is Successful with the Difference of the Intermediate Hop*

```
! Traceroute from BR2-Edge1 to BR3-Edge1:
! The traceroute is successful, but requires ar
intermediary hop,
! identified by the system ip of 10.0.10.1, is
```

```
BR2-Edge1# traceroute vrf 10 10.10.103.1
<<<Omitted for brevity>>>
 1 MPLS-DC1-Edge2 (192.168.20.2) 1 msec 1 msec
 2 BIZ-INET-BR3-Edge1 (209.165.201.8) 1 msec 1
 3 BR3-Edge1-Core (172.16.103.2) 2 msec 3 msec


! Note that no changes have been made to the ce
was
! deployed for Use Case 1, and the number of da
changed.
BR2-Edge1# show sdwan bfd summary
sessions-total 12
sessions-up     12
sessions-max    12
sessions-flap   0
poll-interval  123400
sessions-up-suspended 0
sessions-down-suspended 0
```

It is important to remember that no changes have been made to the centralized control policy applied for Use Case 1. The **show sdwan bfd summary** output in Example 7-4 indicates that there are still 12 BFD sessions (and 12 tunnels) established. This confirms that there are the same

Technet24

number of tunnels as displayed in Figure 7-33, and no new tunnels are established directly between the branches, but the default route is allowing the communication through the data center routers.

Enabling Branch-to-Branch Communication with TLOC Lists

Rather than injecting routes into the Cisco SD-WAN overlay to solve for the desired branch-to-branch communication pattern, it is possible to manipulate the routes that already exist in the overlay to accomplish the same objective.

As discussed in Chapter 3, a TLOC identifier uniquely identifies a SD-WAN tunnel interface in the SD-WAN fabric and also serves as the next-hop attribute that all OMP and network service routes use within the overlay. When routes are advertised by the WAN Edge routers to the SD-WAN Controller (and propagated by the SD-WAN Controller to the other WAN Edge routers), each OMP route is advertised with the local TLOCs as the next-hop values, as illustrated in Figure 7-37. In step 1, all of the WAN Edge routers advertise all of their locally reachable prefixes to the SD-WAN Controller with OMP updates. In step 2, the SD-WAN Controller reflects all these prefixes to all the other WAN Edge routers, using TLOC values as the next-hop addresses.

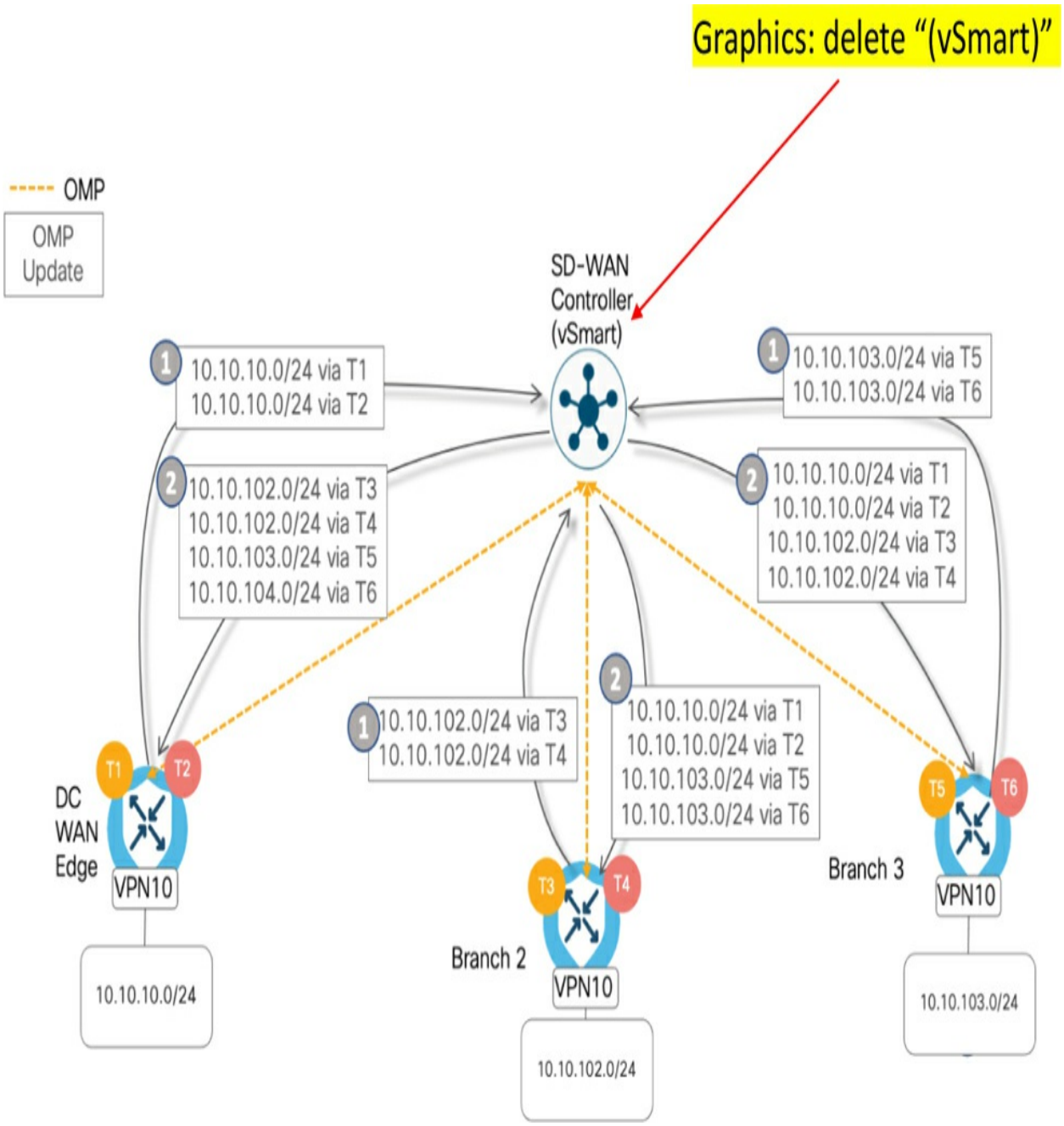


Figure 7-37 *OMP Routes with Next-Hop TLOCs Reflected Through the SD-WAN Controller*



A centralized control policy can be used to manipulate the TLOC attributes

Technet24

that are advertised as part of the OMP routes and, in effect, change the “next-hop address” that the OMP routes recurse to. Through this process, centralized control policies can be a very powerful tool to perform traffic engineering across the SD-WAN fabric. At the same time, this process maintains simplicity by requiring you to apply and execute the policy from only a single location: the SD-WAN Controller.

In order to permit the branch-to-branch communication via the data centers without the need for injecting a default route, in this use case you will slightly modify the policy from Use Case 1 (refer to Example 7-2) so that instead of dropping the routes from the other branches, the routes are instead advertised with new TLOC identifiers: the TLOC identifiers of the DC1 WAN Edge routers. Figure 7-38 illustrates the effect of this policy for the routes advertised from Branch 3.

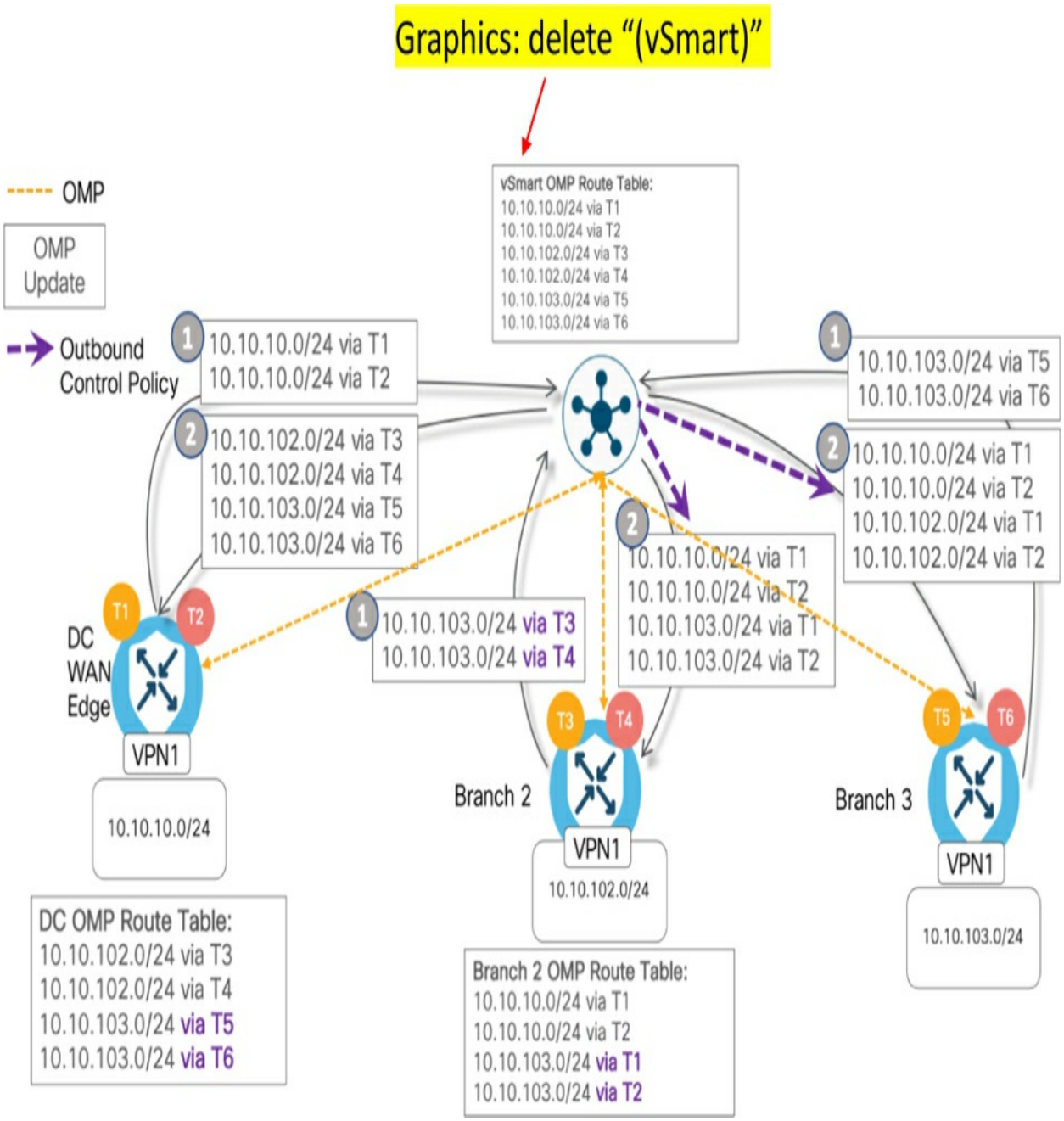


Figure 7-38 *Enabling Branch-to-Branch Communication with TLOC Manipulation*

Figure 7-38 builds on the route exchange process discussed with Figure 7-37. First, all of the WAN Edge routers advertise their locally connected prefixes to the SD-WAN Controllers. In the case of Branch 3, this means the prefix

Technet24

10.10.103.0/24 in VPN 10 is advertised via OMP as being reachable via TLOCs T5 and T6. The result of this advertisement can be seen in the SD-WAN Controller OMP table. Next, this advertisement is reflected to the data center routers as is; no manipulation is performed on the advertisement. However, when the route is reflected to the Branch 2 WAN Edge router, an outbound centralized control policy is applied, and the TLOCs (next-hop addresses) are overwritten from the original values T5 and T6 (the TLOC identifiers of the WAN Edge at Branch 3) to T1 and T2 (the TLOC identifiers of the data center WAN Edge devices).

Note

Throughout the next several pages, we review the detailed process of modifying an existing centralized policy. This section shows the steps you need to take to create a copy of the policies created in the previous use case and then modify them to meet the requirements of this use case. This same process can be used to construct the policies used throughout the other use cases in this book, but configuration steps in the SD-WAN Manager GUI will not be covered to the same degree of detail used for Use Case 1 and Use Case 2.

To begin to construct this policy in the SD-WAN Manager GUI, the first step is to create the new list element that will be needed: a TLOC list. To start this configuration, select the Lists item from the Custom Options menu on the

Configuration > Policies window, as shown in Figure 7-39.

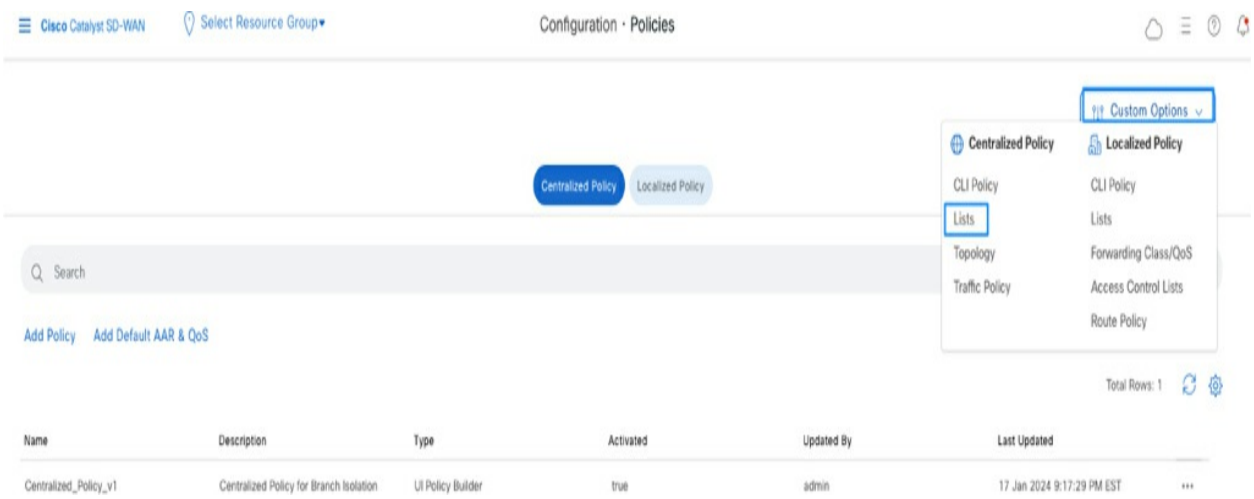


Figure 7-39 *Accessing the List Configuration for Centralized Policies*

To add a new TLOC list, select TLOC from the column of list types on the left and then click the +New TLOC List button, as shown in Figure 7-40.

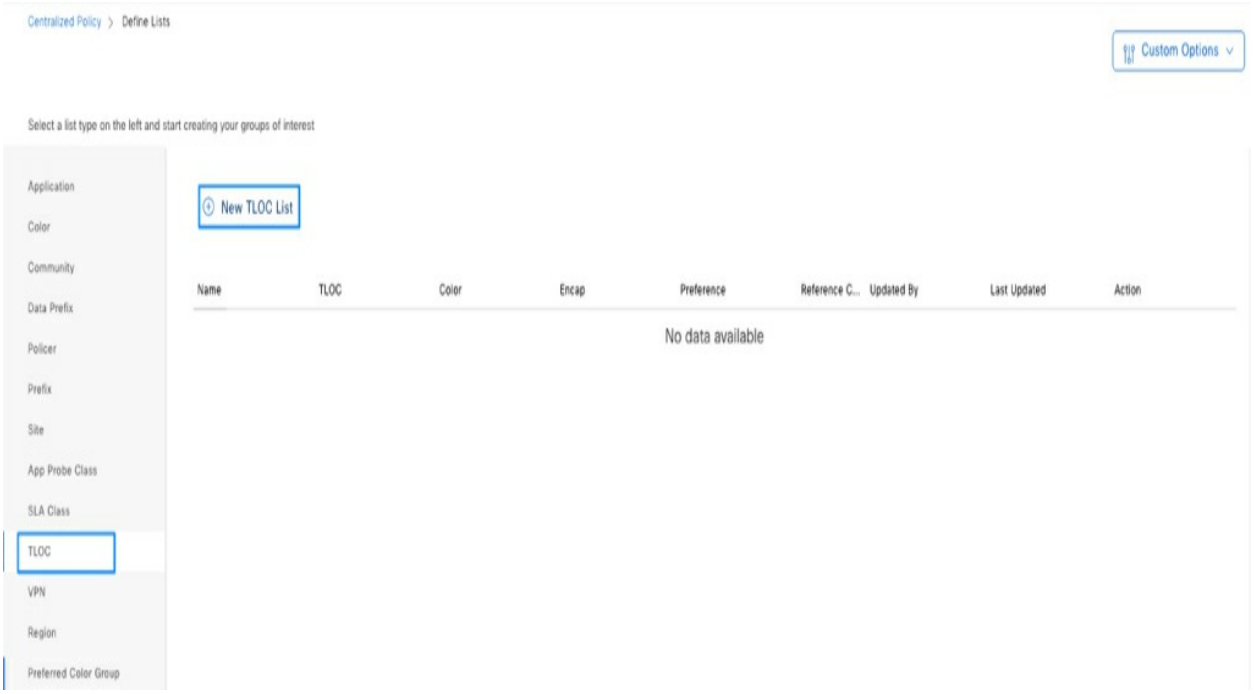


Figure 7-40 *Creating a New TLOC List*

Technet24

Figure 7-41 shows a TLOC list configured with the name DC_TLOCs. There are a total of four TLOCs specified in the TLOC list: the mpls and biz-internet TLOC identifiers on both DC1-Edge1 and DC1-Edge2. Recall that each TLOC identifier is a unique combination of the system IP address, the color, and the encapsulation. In addition to specifying these three required values, an optional preference argument can be used to specify different TLOC preference values. (This optional attribute was not used in this example and will be discussed in later use cases.) Once the list is configured, you can save and close it by clicking the Save button.

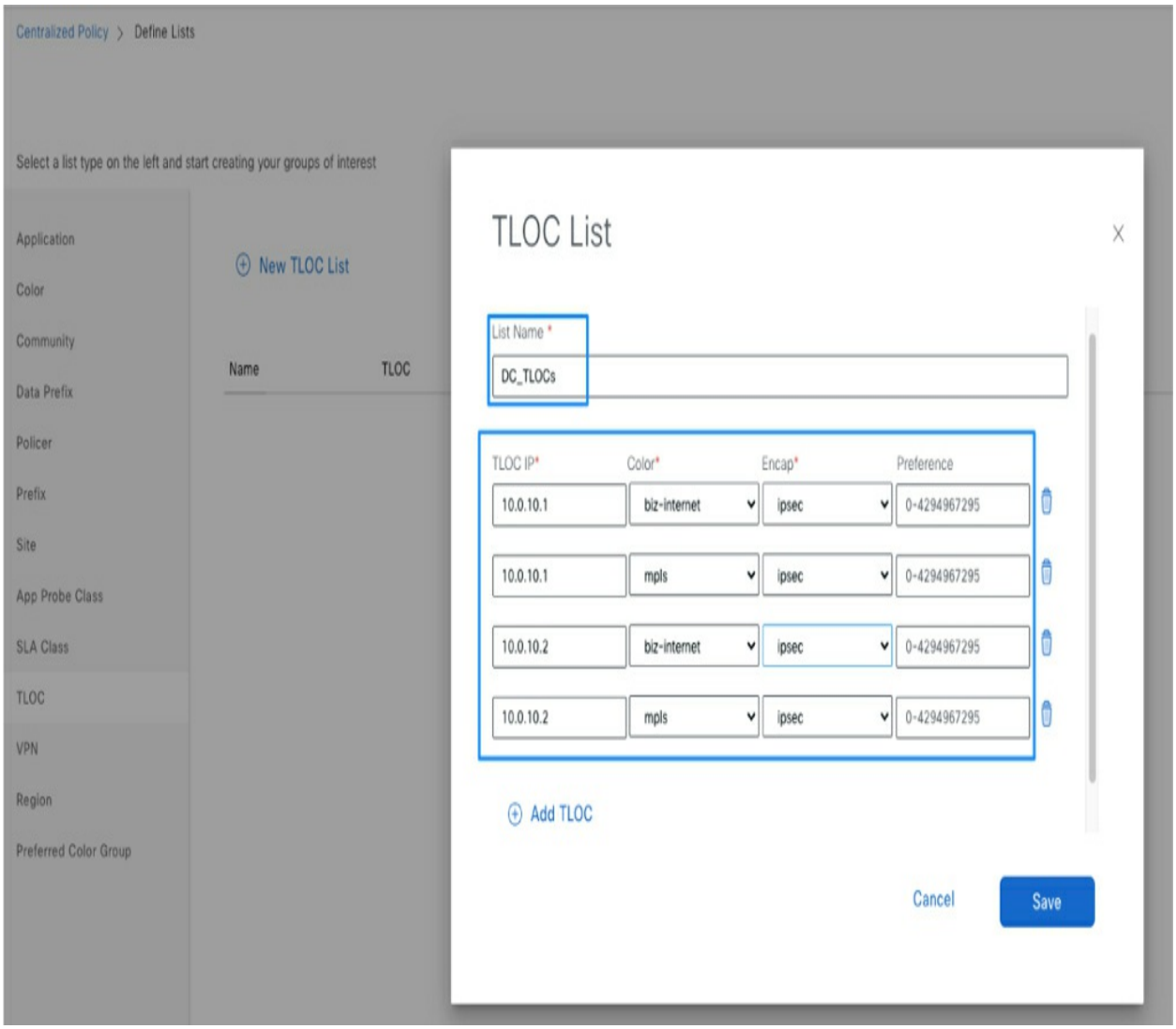


Figure 7-41 *Creating the TLOC List DC_TLOCs*

The next step in the configuration process is to create the necessary centralized control policy to use the TLOC list. In order to do this, select the Topology option from the Custom Options menu in the upper-right corner, as shown in Figure 7-42.

Technet24

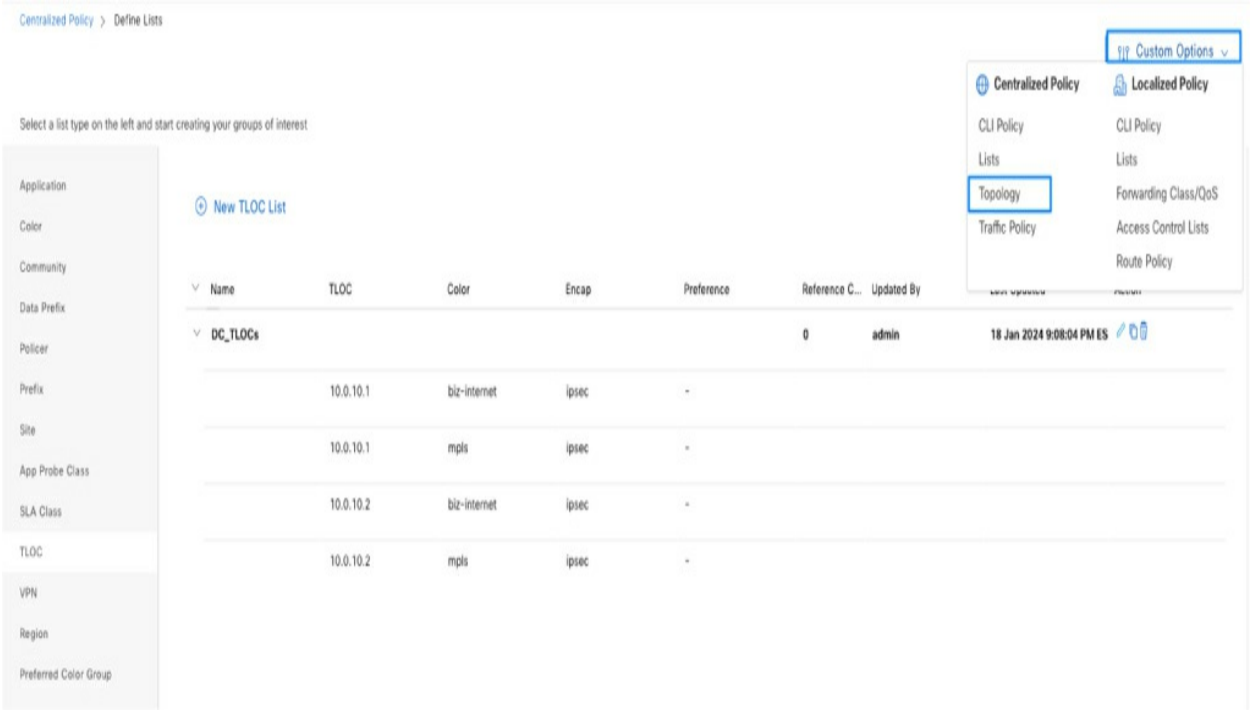


Figure 7-42 *Selecting Topology from the Custom Options Menu*

In order to copy the existing policy that was created in Use Case 1 so that it can be modified, click the three dots (…) menu next to the policy and select Copy, as shown in Figure 7-43.

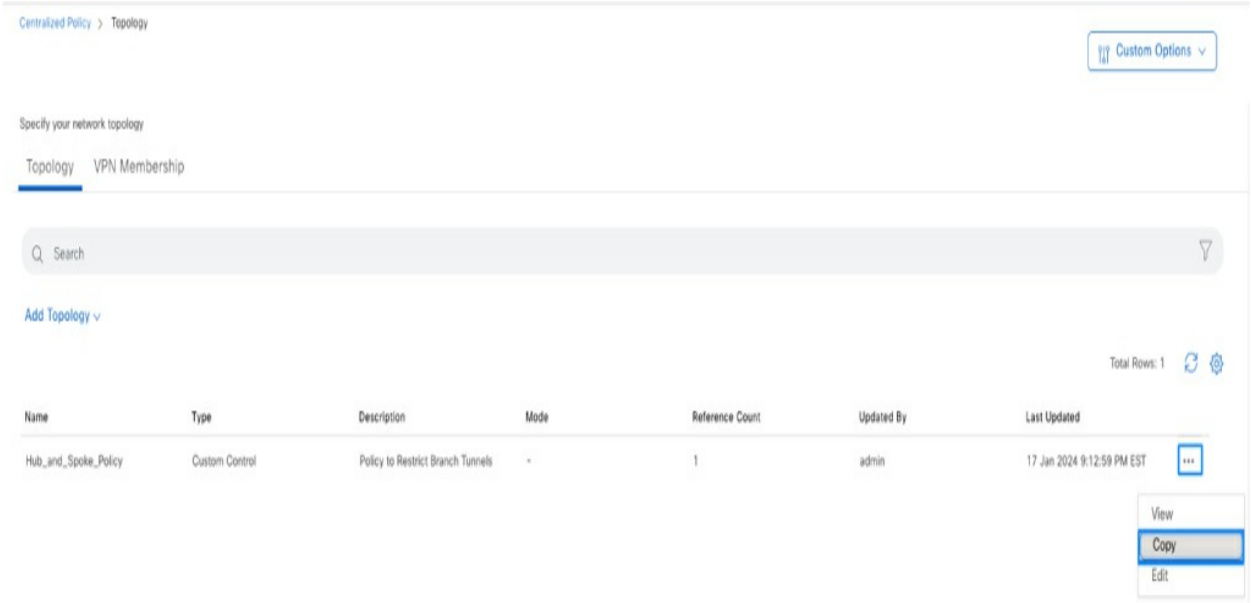


Figure 7-43 *Creating a Copy of an Existing Centralized Control Policy*

Every policy is required to be configured with a name and a description. In this case, the new centralized control policy is named Hub_and_Spoke_TLOC_Lists, as shown in Figure 7-44.

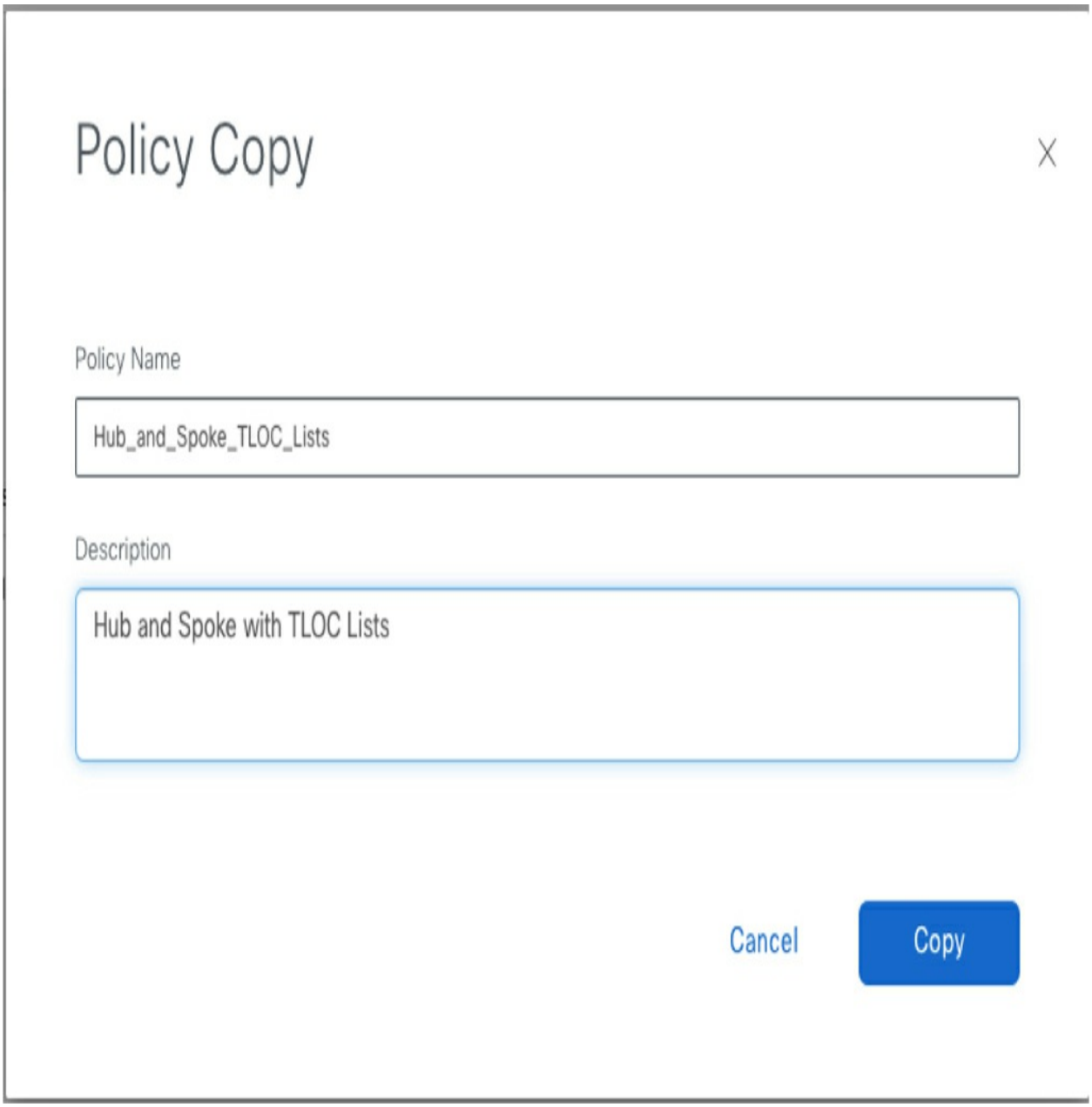


Figure 7-44 *Providing a Name and a Description for the Newly Copied*

Technet24

Policy

Now that the new copy of centralized control policy has been created from the original policy in Use Case 1, you can edit the policy by selecting the Edit option from the three dots menu, as shown in Figure 7-45.

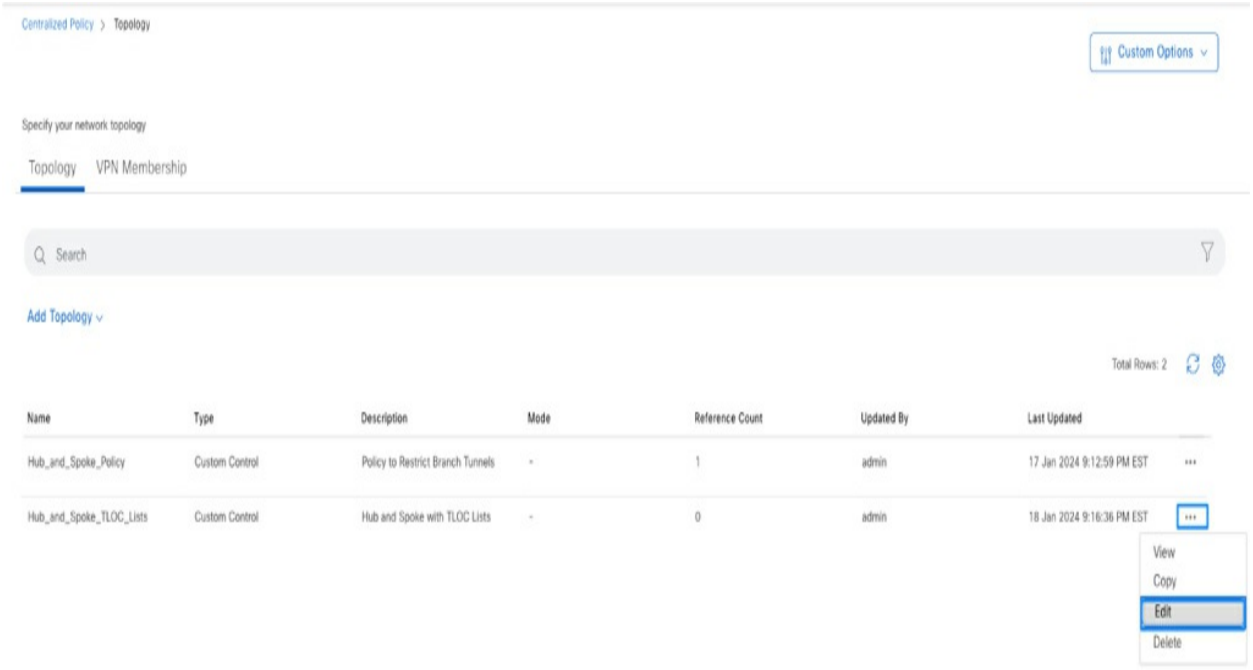


Figure 7-45 *Editing the Newly Copied Centralized Control Policy*

This policy needs to be altered so that the routes that are originating from the branch sites are rewritten to resolve to the new DC_TLOCs list that was previously created (see Figure 7-46). In order to accomplish this, you need to select the existing Route sequence type on the far left in the GUI. Then, click +Sequence Rule to add a new sequence rule. In this sequence rule, you configure the matching criterion as the BranchOffices site list that was created in the previous use case by clicking the Match sub-tab, selecting the Site criterion, and adding the necessary site list (not pictured). Once the

matching criterion is specified, select the Actions tab at the top of the sequence rule and select the Accept radio button. Once the routes are accepted, you can select the TLOC action and specify the previously created TLOC list DC_TLOCs. When the configuration is completed, select the blue Save Match and Actions button at the bottom of the sequence rule.

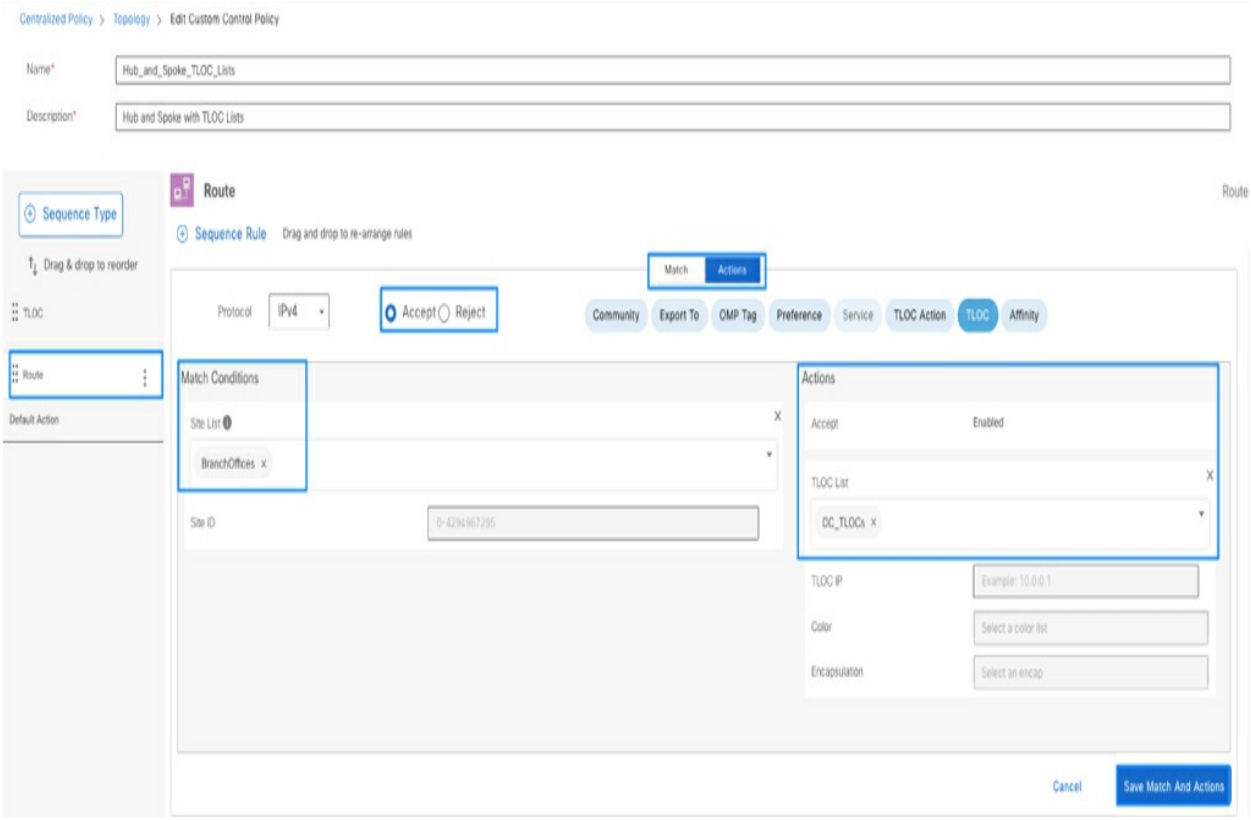


Figure 7-46 *Creating a New Sequence to Manipulate Routes Advertised from Branch Sites*

The completed policy with both the existing route sequence rule from the previous use case and the newly created route sequence rule are shown in Figure 7-47. In summary, the first sequence rule of this policy will match all the routes being advertised from the data centers and forward them on

Technet24

without any modifications. The second sequence will match all the routes being advertised from other branches and change their TLOC identifiers (or next-hop attributes) to be the TLOC identifiers of the data centers. You save the completed policy by clicking the blue Save Control Policy button at the bottom.

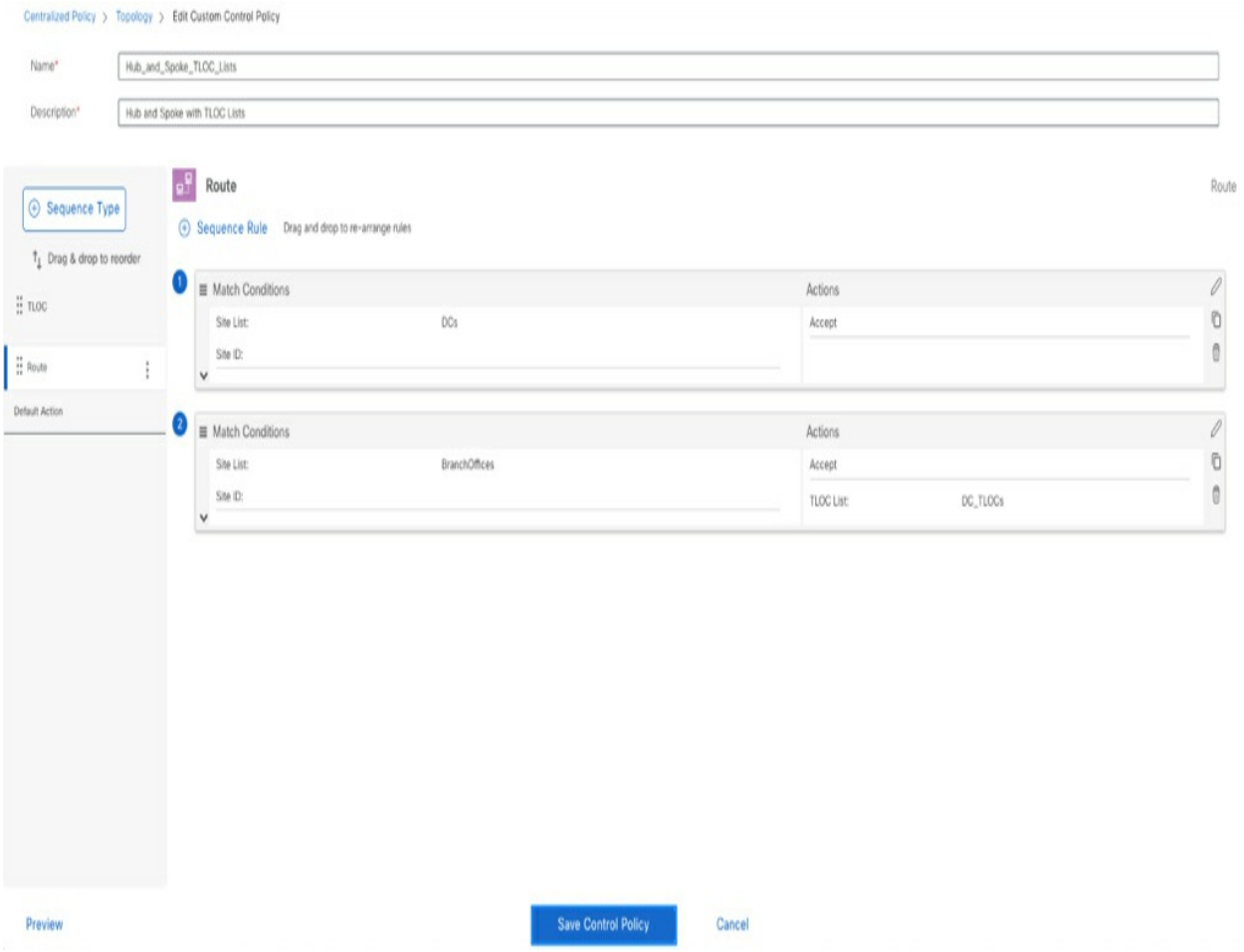


Figure 7-47 *Saving the Centralized Control Policy with Two Route Sequences*

Now that the centralized control policy has been created, it needs to be imported into a centralized policy. To do this, return to the main Centralized

Policy Configuration screen by clicking Centralized Policy in the menu option at the top of the screen, as shown in Figure 7-48.

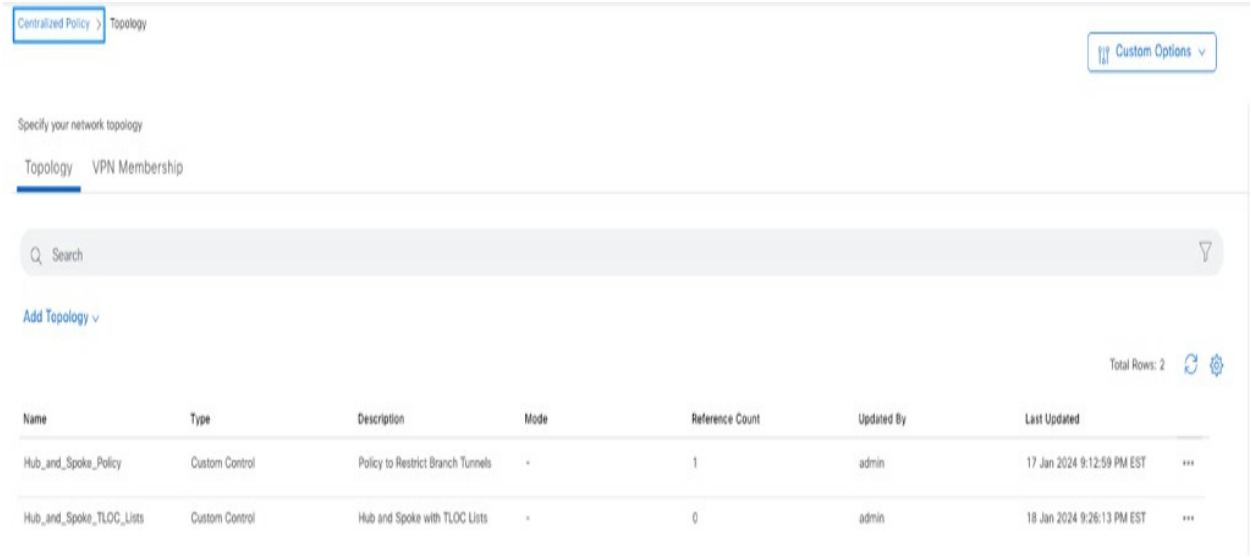


Figure 7-48 *Navigating Back to the Centralized Policy Configuration Screen Using the Breadcrumb Trail*

The next step in the configuration process is to create a copy of the centralized policy that was created in Use Case 1, which will then be modified. As shown in Figure 7-49, do this by clicking the three dots menu and selecting the Copy option.

Technet24

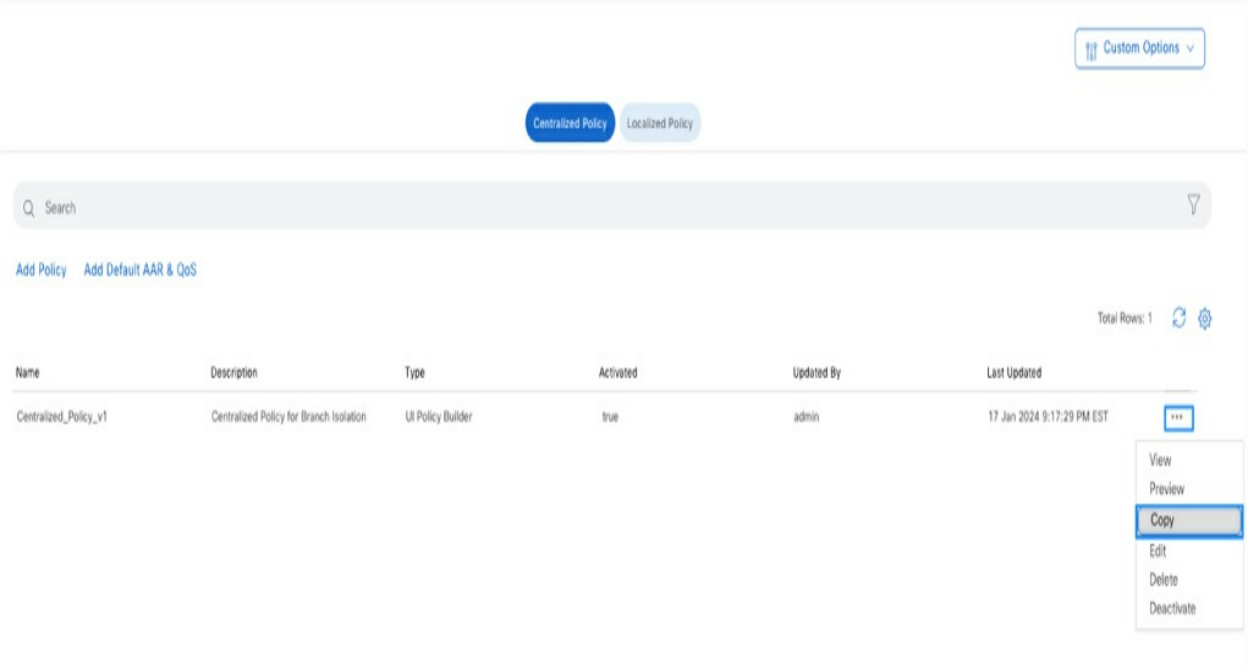


Figure 7-49 *Creating a Copy of a Centralized Policy Before Starting Modifications*

In order to copy the policy, you need to supply a name and description, as shown in Figure 7-50. Once these values are supplied, click the Copy button to complete the process.

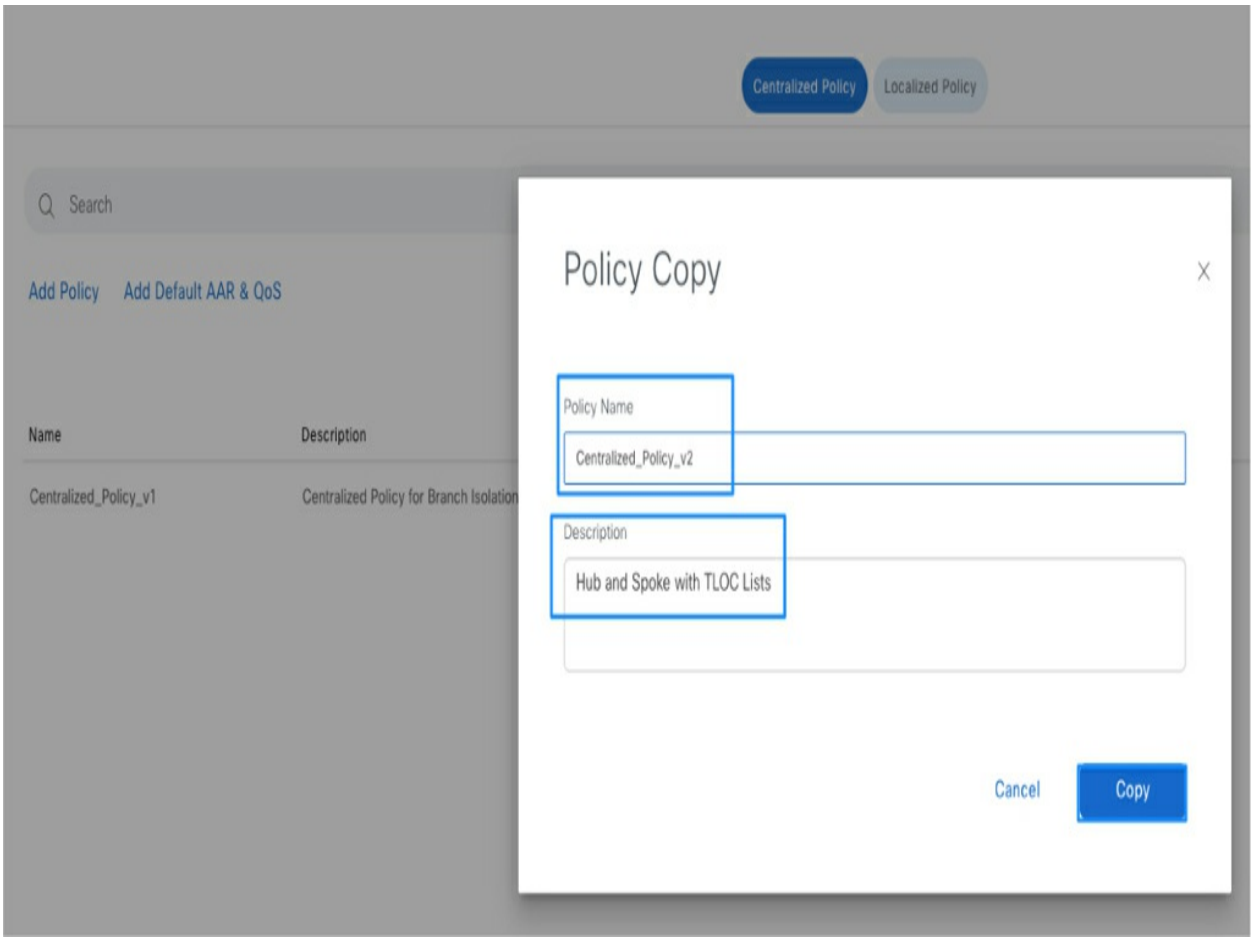


Figure 7-50 *Configuring a Name and Description for the Copied Centralized Policy*

Figures 7-49 and 7-50 show the rest of the process for copying the centralized policy, which is similar to the steps taken in Figures 7-43 and 7-44 for copying the centralized control policy. In both cases, it would have been possible to modify the existing policies rather than copy them and modify the copies. However, the process illustrated here is considered a best practice as it helps you avoid inadvertently applying a configuration to the network and aids in quickly rolling back any configuration changes should the need arise.

Now that you have created Centralized_Policy_v2, you can modify it to use

Technet24

the new control policy with the TLOC list by selecting the Edit option from the three dots menu, as shown in Figure 7-51.

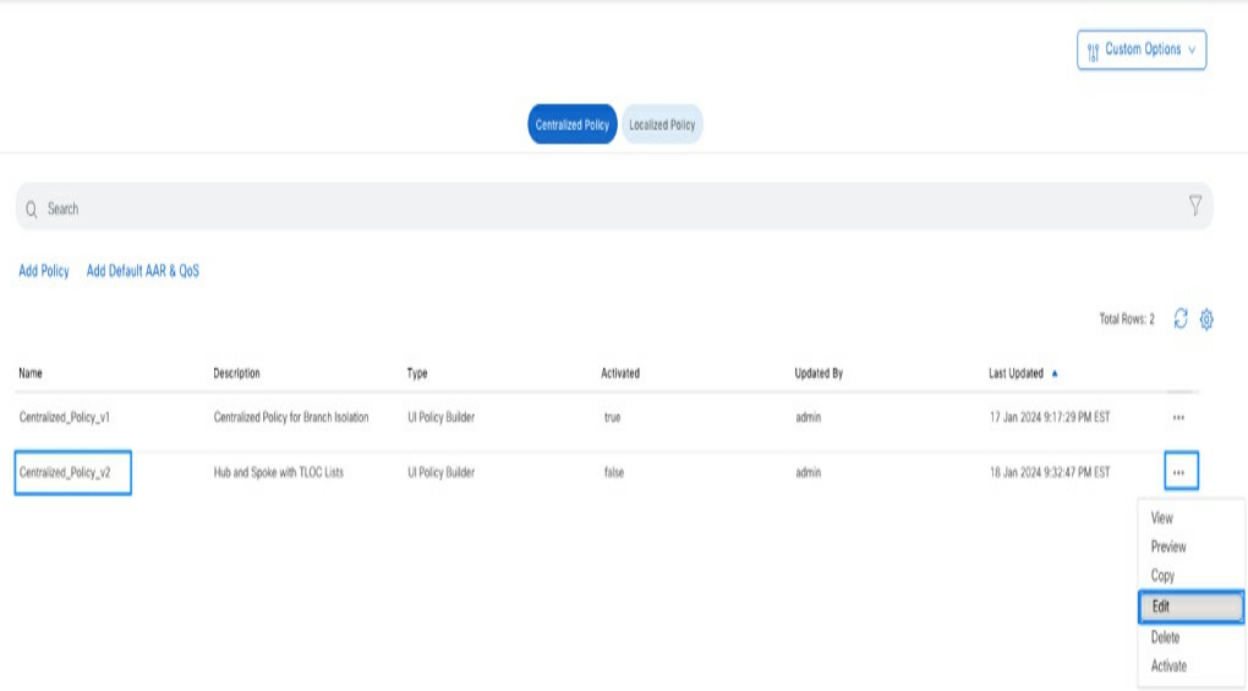


Figure 7-51 *Editing a Centralized Policy*

In order to modify this centralized policy for Use Case 2, you need to make several changes. The centralized control policy created in Use Case 1 needs to be separated from this policy, and the new centralized control policy that was created in Figures 7-43 through 7-47 needs to be imported into this centralized policy. Finally, that new policy has to be applied to the branch sites. In order to separate the old centralized control policy, click on the Topology tab at the top of the window, as shown in Figure 7-52. On the Topology tab, the Topology sub-tab is already selected, and the centralized control policies that are referenced by this centralized policy are listed. (Note that there is also a VPN Membership sub-tab that you will use in later use

cases.) Select the Detach option from the three dots menu on the Hub_and_Spoke_Policy in order to separate this control policy from Use Case 1.

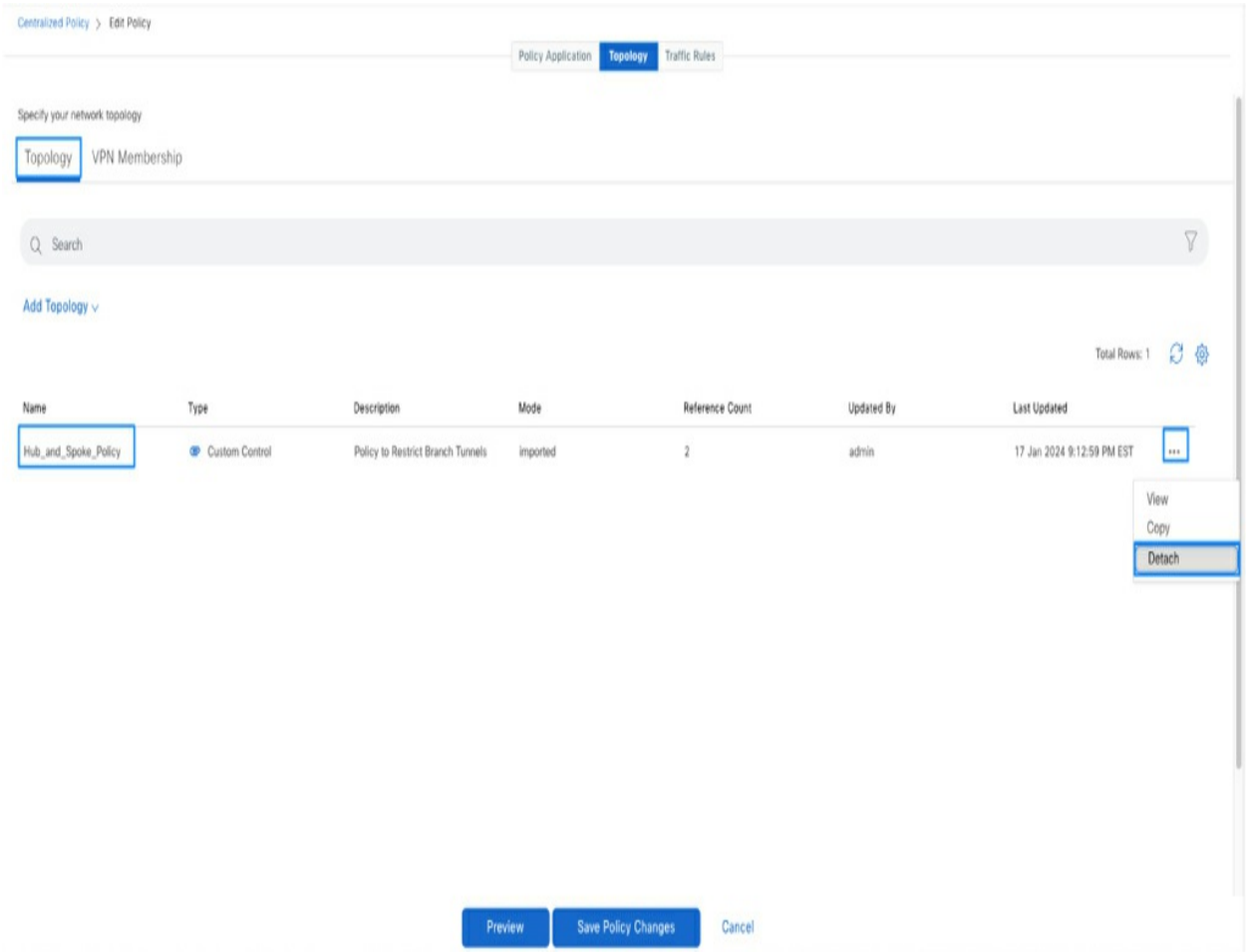


Figure 7-52 *Separating an Unnecessary Centralized Control Policy from a Centralized Policy*

Next, you need to attach the newly created centralized control policy to this policy by clicking the +Add Topology button and selecting the Import Existing Topology option, as shown in Figure 7-53.

Technet24

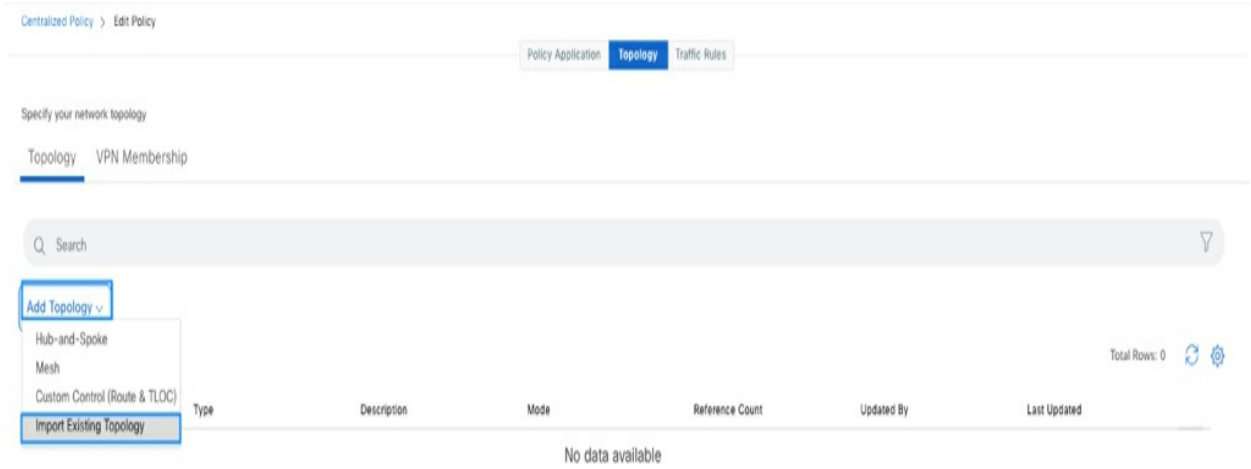


Figure 7-53 *Importing an Existing Control Policy into a Centralized Policy*

In the Import Existing Topology dialog box, select the radio button for Custom Control (Route and TLOC) under Policy Type and then select the name of the new centralized control policy Hub_and_Spoke_TLOC_Lists from the Policy drop-down. Complete the process by clicking the blue Import button, as shown in Figure 7-54.

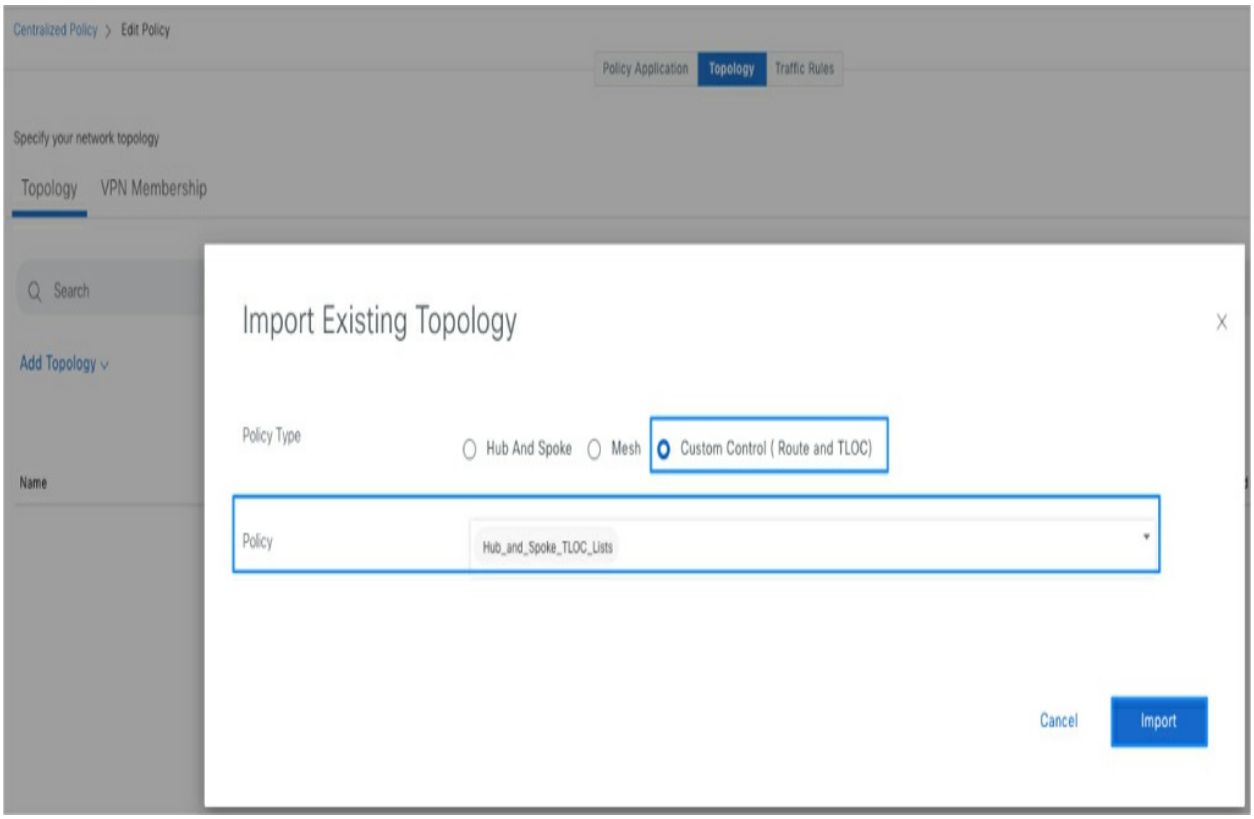


Figure 7-54 *Selecting the Existing Control Policy to Import into a Centralized Policy*

Now that the new centralized control policy has been imported into the centralized policy, the last part of the configuration process is to specify where this policy should be applied. To start this process, move back to the Policy Application page by clicking that tab at the top of the window. The Topology sub-tab, which is selected by default, lists all the control policies in this centralized policy. (You will use the other sub-tabs in later chapters with application-aware routing and centralized data policies.)

Under the Hub_and_Spoke_TLOC_Lists policy, click the blue +New Site List button in order to specify the site lists that this policy should be applied

Technet24

to. Add the BranchOffices site list to the Outbound Site List field. The final step in saving the policy application configuration is to click the Add button on the right side and then click the Save Policy Changes button at the bottom, as shown in Figure 7-55.

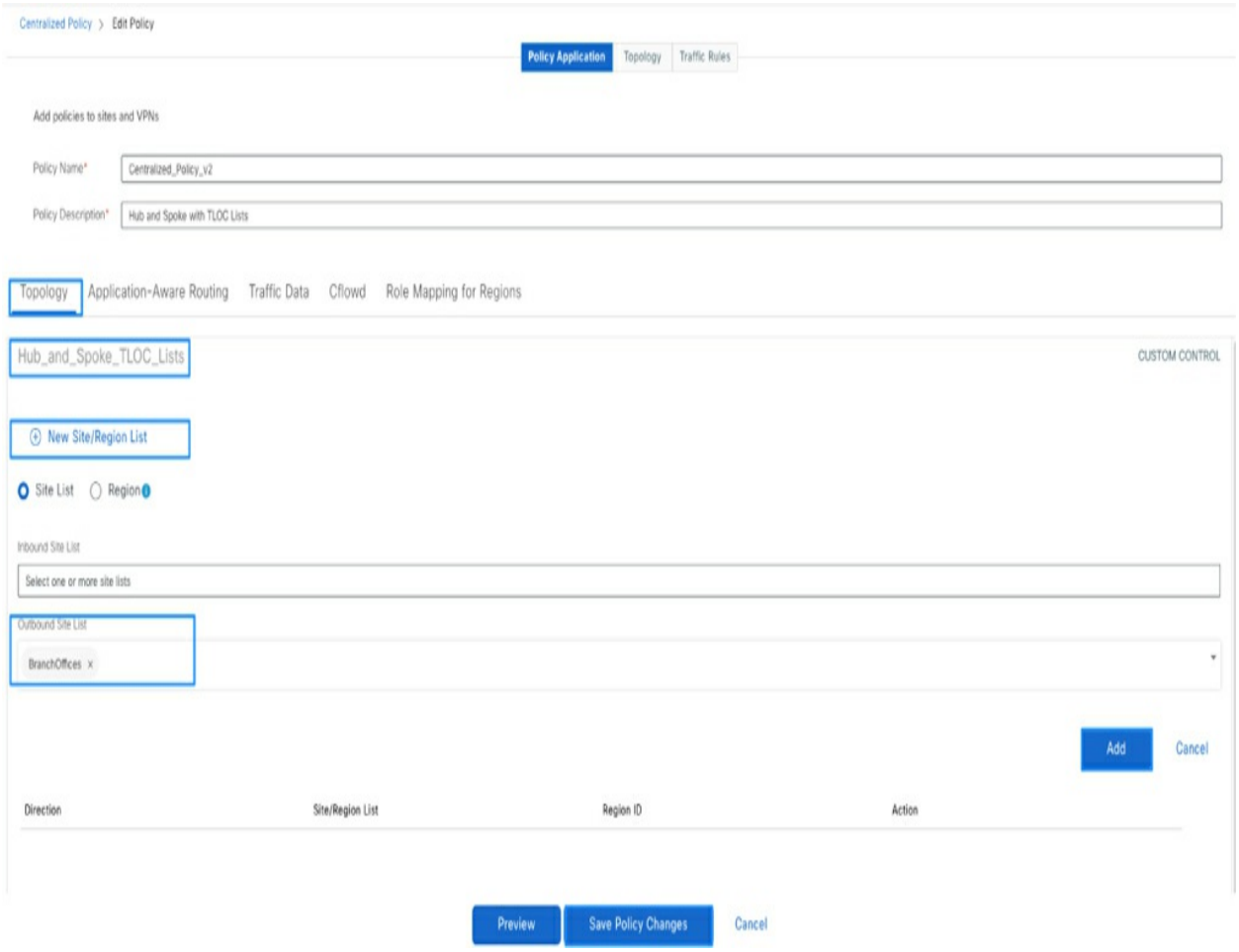


Figure 7-55 *Configuring the Policy Application for the Imported Control Policy in a Centralized Policy*

The new centralized control policy has now been saved, and you can activate it by selecting the Activate option from the three dots menu, as shown in Figure 7-56. There is no need to deactivate Centralized_Policy_v1 before

activating the new policy. Because there can only be a single centralized policy that is active at any point in time, activating the new policy automatically deactivates any other policy.

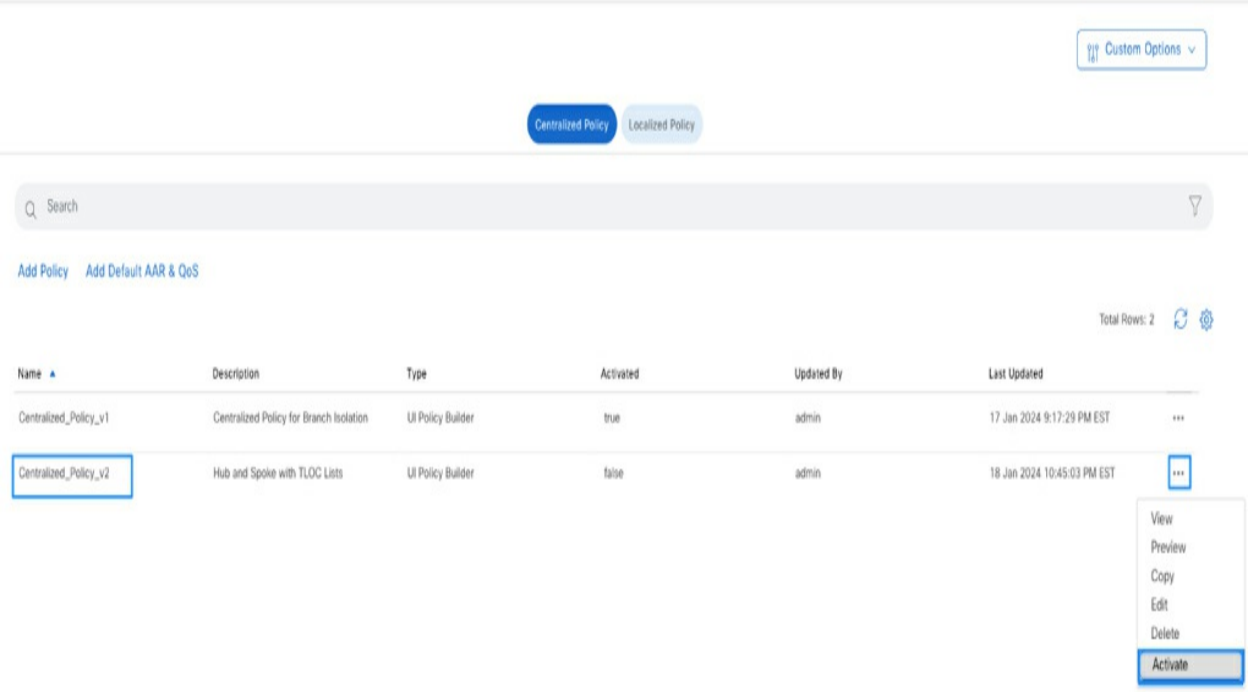


Figure 7-56 *Activating a Centralized Policy*

You can see the full CLI configuration of the policy by selecting the Preview option from the three dots menu (refer to Figure 7-56). The output of the policy preview displayed in Example 7-5 shows the complete centralized policy used to implement the objectives of this use case.

Example 7-5 *Use Case 2: Hub-and-Spoke Policy with TLOC Lists*

```
! Much of the centralized policy is unchanged
```

Technet24

```
! The relevant changes in sequence 21 and the l
policy
 control-policy Hub_and_Spoke_TLOC_Lists
    sequence 1
     match tloc
      site-list DCs
     !
     action accept
     !
    !
    sequence 11
     match route
      site-list DCs
      prefix-list _AnyIpv4PrefixList
     !
     action accept
     !
    !
! The new sequence 21 has been added to permit
! advertised from the branches, and advertise t
! specified by the "DC_TLOCs" argument to the t
   sequence 21
    match route
     site-list BranchOffices
     prefix-list _AnyIpv4PrefixList
```

```
      !
      action accept
       set
        tloc-list DC_TLOCs
       !
      !
     !
  default-action reject
 !
 lists
  site-list BranchOffices
   site-id 100-199
  !
  site-list DCs
   site-id 10-50
  !
! A new list called "DC_TLOCs" is used to spec
! advertised as the next hop addresses of the
  tloc-list DC_TLOCs
   tloc 10.0.10.1 color mpls encap ipsec
   tloc 10.0.10.1 color biz-internet encap ips
   tloc 10.0.10.2 color mpls encap ipsec
   tloc 10.0.10.2 color biz-internet encap ips
  !
  prefix-list _AnyIpv4PrefixList
```

Technet24

```
    ip-prefix 0.0.0.0/0 le 32
   !
  !
!
apply-policy
 site-list BranchOffices
  control-policy Hub_and_Spoke_TLOC_Lists out
 !
!
```

Example 7-5 shows the command-line configuration for the TLOC list that configured in Figure 7-41. The TLOC list is established with the command **tloc-list** {*list-name*}. Once the list is created, each TLOC much be specified individually, using the following syntax:

```
tloc {system-ip} color {color} encap {ipsec|gre}
[preference preference]
```

These configuration commands reflect the configuration performed in SD-WAN Manager in Figure 7-41. The TLOC list is then referenced in Sequence 21 of the control policy to overwrite the existing TLOC attributes that would have been advertised with the OMP routes from the branch sites. Finally, the last stanza in Example 7-5 shows that the policy is applied to the site list BranchOffices in the outbound (**out**) direction.

You can see the effect of applying this new centralized policy by looking at the routing table, as shown in Figure 7-57. In the IP routing table, you can see that you now have four different paths to the prefix 10.10.103.0/24. Careful inspection of those advertisements reveals that the TLOC IP address is not 10.0.103.1, as it was in Figure 7-4, with no control policy applied. Instead, it is now listed as 10.0.10.1 and 10.0.10.2, the TLOC IP addresses of the WAN Edge routers in DC1, as specified in the TLOC list.

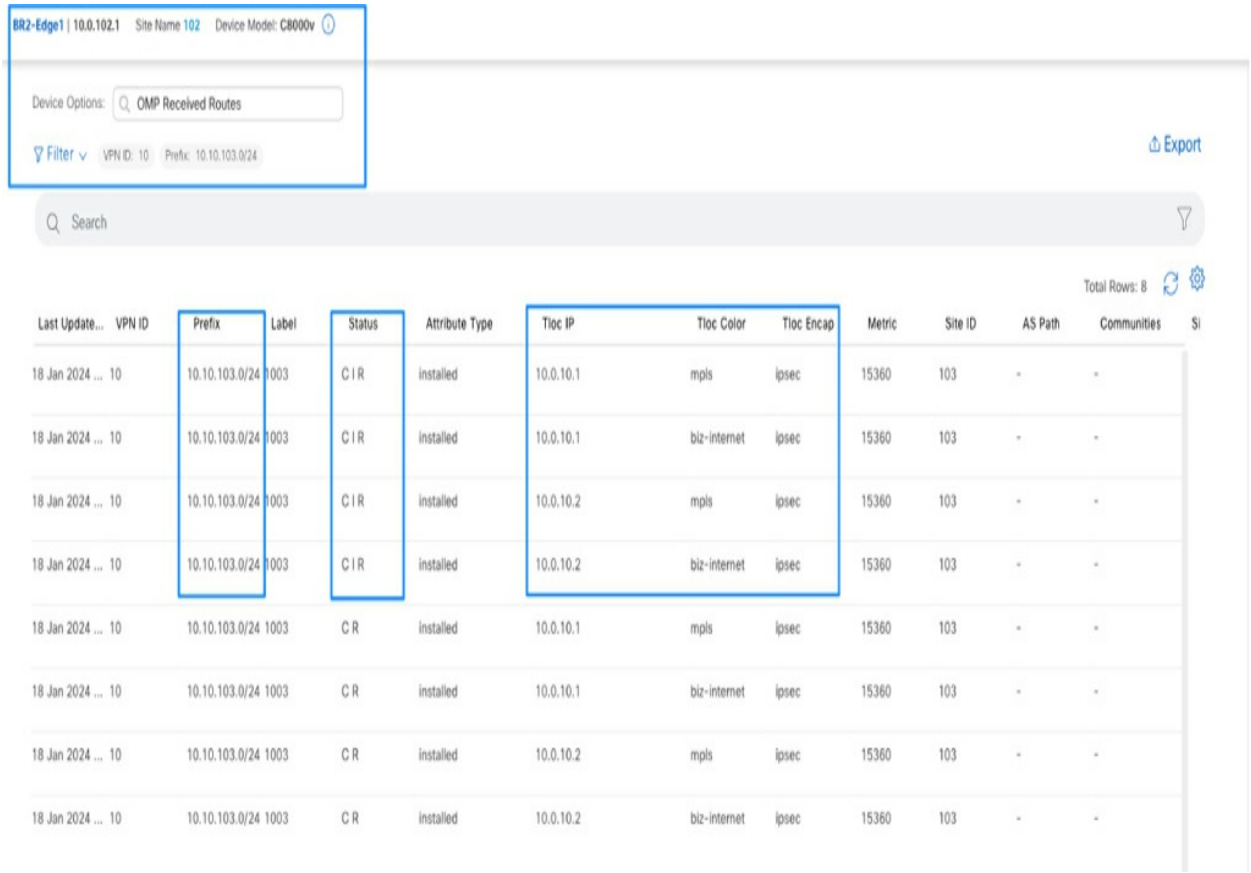


Figure 7-57 *Routing Table on BR2-Edge1*

It is also important to consider what effect the policy had on the routes that have been advertised to the data center. You can examine the IP routes output

Technet24

in the Real Time view, as shown in Figure 7-58, and see that the routes are received by the DC1-Edge1 router with all the original TLOC advertisements.

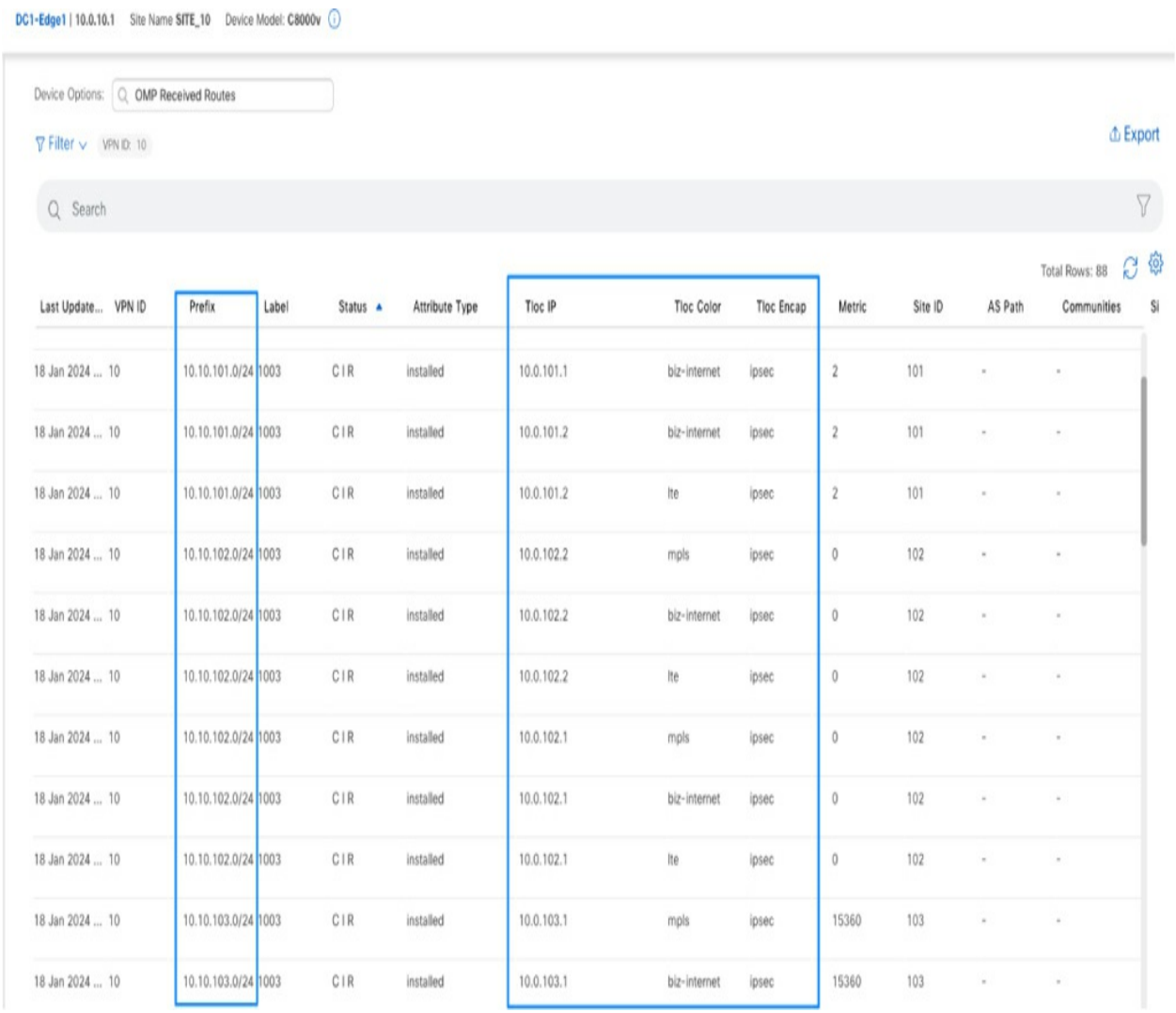


Figure 7-58 *Routing Table on DC1-Edge1*

As shown in Figure 7-58, the data center routing table entries for the prefixes from the branches are specified with the original TLOC IP addresses as the next-hop values. When you compare Figure 7-58 to Figure 7-57, you can clearly see that there are different TLOC IP addresses, or next-hop attributes,

that are advertised to the branch WAN Edge devices than are advertised to the data center WAN Edge devices. This is expected because the route manipulation is happening *outbound* to the WAN Edges covered under the BranchOffices site list. There is no manipulation happening on the OMP advertisements being sent to the data center WAN Edge routers. Furthermore, a **traceroute** from the Branch 2 WAN Edge router is able to confirm that the data plane is working as expected, as shown in Example 7-6.

Example 7-6 *BR2-Edge1 Is Able to Access the Prefixes in Another Branch by Transiting the Data Center*

```
! BR2-Edge1 is able to reach the destination 10
requires
! additional hops as it must transit through th
IP 10.0.10.1).
BR2-Edge1# traceroute vrf 10 10.10.103.1
<<<Output omitted for brevity>>>
  1 MPLS-DC1-Edge2 (192.168.20.2) 1 msec 2 msec
  2 BIZ-INET-BR3-Edge1 (209.165.201.8) 1 msec 1
  3 BR3-Edge1-Core (172.16.103.2) 2 msec 3 msec
BR2-Edge1#
```

Use Case 2 Review

Technet24

This use case describes two different mechanisms to permit branch-to-branch communication in an SD-WAN fabric where TLOC routes were filtered to prohibit the establishment of direct branch-to-branch data plane tunnels. In the first mechanism, you advertised summary prefixes from the data center in order to draw branch-to-branch traffic, for which the branch routers did not have more specific routes, to the data centers. The data centers, in turn, could then use their more specific routing information to forward the traffic on to its final destination.

In the second mechanism, which is based on using a TLOC list, you used the **set tloc-list** action in the centralized control policy to manipulate the OMP routes so that the prefixes from remote branches were advertised with the TLOC identifiers of the data centers. Since the WAN Edge routers had already received the DC TLOC advertisements and had built tunnels to the data center WAN Edge routers, the OMP routes with these modified next-hop addresses could be used to forward traffic.

Note

A TLOC list is an incredibly powerful tool to create flexible policies in order to implement traffic engineering and meet the objectives of the network administrator. But remember that with great power comes great responsibility. Improper application of TLOC lists can cause unintended and unexpected forwarding behaviors. In addition, a TLOC list

requires a static definition in the centralized control policy and is not updated automatically in the same manner as traditional OMP routing updates. For example, if (years after implementing this policy) an additional transport link is added to the data center and an additional tunnel interface is added to DC1-Edge1 and DC1-Edge2, the TLOC list used for this traffic engineering policy would not change automatically. The network administrator would need to remember to update the TLOC list if using all three transport networks is the desired behavior. In a more extreme example, if the WAN Edge routers at DC1 were to be assigned new system IP addresses, the addresses in the TLOC list would not automatically be updated. Traffic flows from branch to branch would then fail, even as the traffic flows from branch to DC would succeed.

Therefore, we recommend using TLOC lists only when they are the only tool able to accomplish the desired outcome. When considering the objective in this use case—that is, enabling branch-to-branch communication without building branch-to-branch tunnels—we strongly recommend using network summarization rather than the TLOC list method.

Use Case 3: Traffic Engineering at Sites with Multiple Routers

Technet24

Currently in the sample network, three different sites are deployed with redundant active routers: Data Center 1, Data Center 2, and Branch 1. The default behavior when forwarding traffic across the fabric is to load-share across all equal paths, and in the case of these three sites, that could mean load-sharing across all of the paths from both routers, as shown in Figure 7-59.

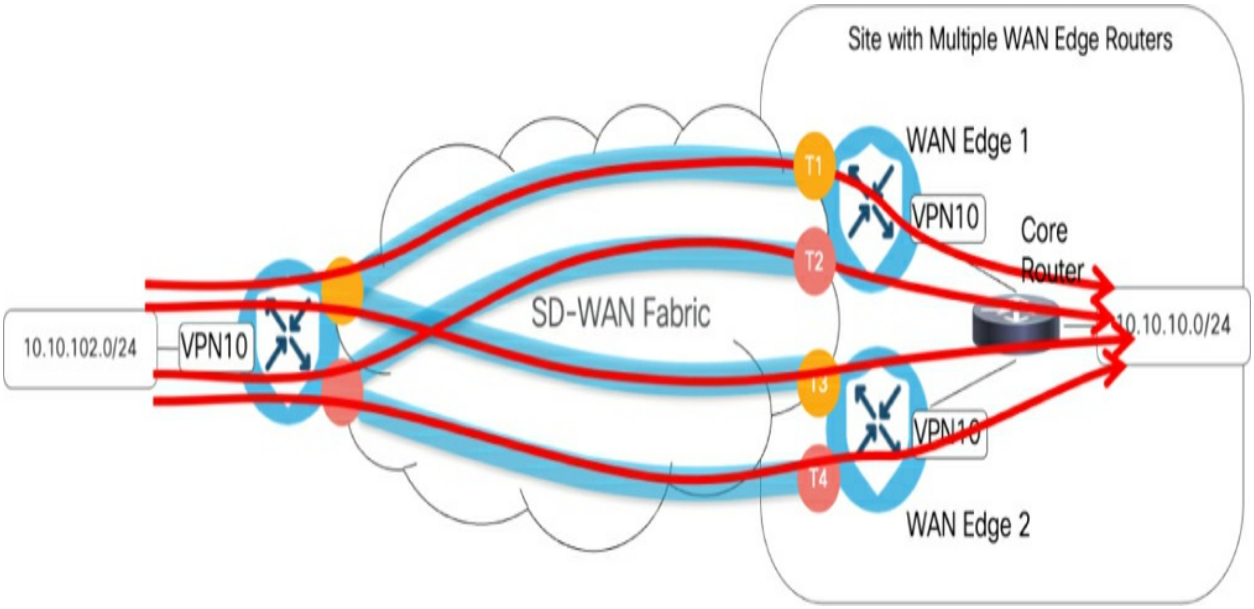


Figure 7-59 *Four Paths to 10.10.10.0/24*

In Figure 7-59, traffic that is destined to the prefix 10.10.10.0/24 may take any of the four established tunnels across the SD-WAN fabric. However, certain issues can arise if a device outside the SD-WAN fabric, such as the core router in Figure 7-60, forwards the return flow across a WAN Edge node other than the router on which the original flow was received. Because the SD-WAN fabric itself is stateless, there is no inherent problem with using multiple routers for flows to and from a specific destination. However, some

advanced data plane services, such as deep packet inspection or the on-box security capabilities, require the ability to see both sides of a traffic flow in order to provide optimal application layer services.

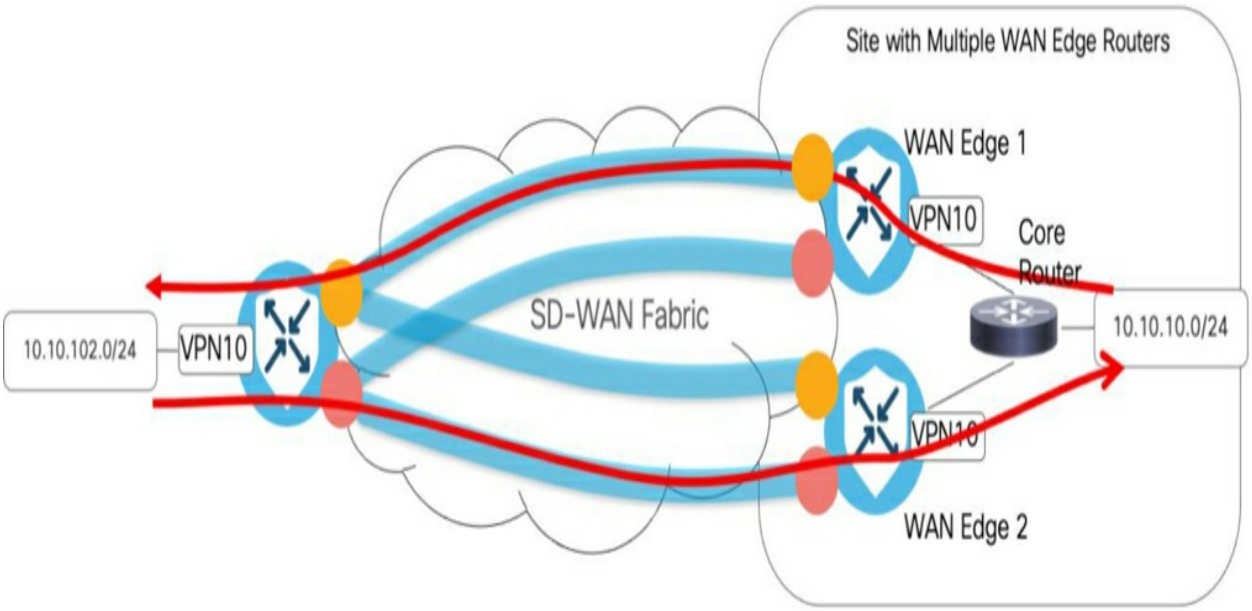


Figure 7-60 *Multipathing on the Fabric May Use Multiple WAN Edge Routers for a Single Flow*

In order to ensure that both halves of a flow are transiting the same WAN Edge router at a given site, it is common to configure the fabric so that all of the traffic traverses one of the routers in steady state, and the other router is not actively passing traffic until a failure occurs.

Note

Completing this configuration involves both how traffic is forwarded across the SD-WAN fabric to the WAN Edge devices and how traffic is forwarded back from the LAN

Technet24

switches to the WAN Edge devices. Chapter 6 discusses these separate and distinct policy domains.

This use case focuses on manipulating the SD-WAN fabric to prefer a single WAN Edge router when traffic is coming in from the fabric. The corresponding configuration for preferring a single WAN Edge router from the LAN side is discussed in detail in Chapter 10, "Localized Policies."

As discussed in Chapter 3, the Preference attribute for both TLOCs and OMP routes is evaluated in the OMP best-path selection process. Preference can therefore be altered to manipulate the OMP path selection. In many areas of networking, there is more than one way to accomplish any given objective. This use case explores two different methods for manipulating the TLOC Preference values in order to complete this traffic engineering use case. In the following sections, the manipulations at the data centers will be completed with a centralized control policy, and changes at Branch 1 will be made in the WAN Edge configurations locally.

Setting TLOC Preference with Centralized Policy

With the existing policy from Use Case 2 (refer to Example 7-5), both DC1-Edge1 and DC1-Edge2 are advertising equal-cost paths to the 10.10.10.0/24 prefix in DC1, as shown in Figure 7-61.

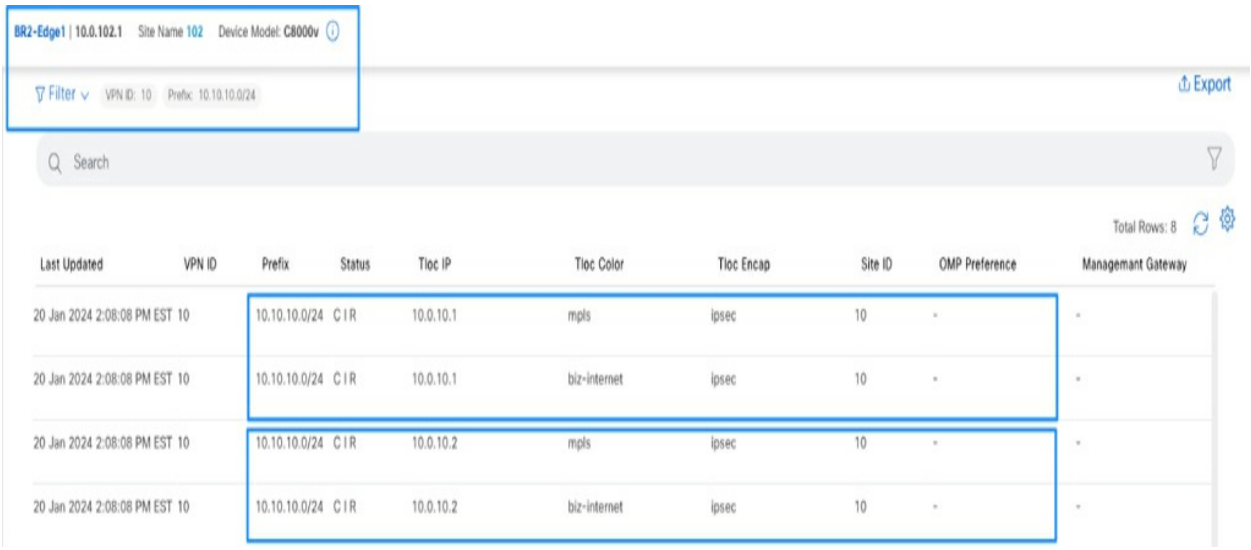


Figure 7-61 *Four Paths to 10.10.10.0/24 Transiting Two Different WAN Edge Routers*

There are a total of four equal-cost paths for the prefix 10.10.10.0/24: two from DC1-Edge1 and two from DC1-Edge2. Each WAN Edge device is advertising a path with the colors mpls and biz-internet.

In the previous use cases, you have seen how to use centralized control policies applied in the outbound direction in order to filter and manipulate routes advertised from the SD-WAN Controller. Because the outbound manipulation does not affect the OMP routes and TLOCs in the SD-WAN Controller's own tables, they can be more limited in scope, making it easy to apply different changes in the advertisements to different sites. You saw this in the previous use case when the TLOC values for certain OMP routes from the branches were overwritten when advertised to some sites (branches) but not other sites (data centers). Conversely, when the intent is to make a global change to OMP routes or TLOC routes, it is often more appropriate to use an

Technet24

inbound centralized control policy. Inbound centralized control policies make manipulations before the SD-WAN Controller best-path selection algorithm is applied and before the routes are inserted into the SD-WAN Controller table. The manipulations made by inbound control policies are apparent in the advertisements to all other OMP peers unless overwritten by an additional outbound control policy.

Because the intention with this use case is to have all of the routers prefer to send their traffic to Edge1 over Edge2 in the data centers, you will use an inbound centralized policy to achieve this global change. Example 7-7 shows the changes to the centralized control policy that are required to achieve this objective.

Note

Parts of the policy in Example 7-7 that are either irrelevant to this use case or that are unchanged from the previous use case are omitted for clarity and brevity.

Example 7-7 *Use Case 3: Centralized Policy Setting TLOC Preference*

```
policy
 control-policy Hub_and_Spoke_TLOC_Lists
 ! <<<No changes made to this policy from Exam
```

```
 !
! A new control policy is created to set the TL
 control-policy Set_DC_TLOC_Preference
    sequence 1
     match tloc
      originator 10.0.10.1
     !
     action accept
      set
       preference 500
      !
     !
    !
    sequence 11
     match tloc
      originator 10.0.10.2
     !
     action accept
      set
       preference 400
      !
     !
    !
    sequence 21
     match tloc
```

Technet24

```
        originator 10.0.20.1
       !
       action accept
        set
         preference 500
        !
       !
      !
      sequence 31
       match tloc
        originator 10.0.20.2
       !
       action accept
        set
         preference 400
        !
       !
      !
   default-action accept
  !
  lists
  ! <<<No changes made to the lists from Example
  !
!
! The apply-policy statement has been modified
```

```
! should be applied inbound on advertisements r
apply-policy
 site-list DCs
  control-policy Set_DC_TLOC_Preference in
 !
 ! The existing policy for the branches remains
 site-list BranchOffices
  control-policy Hub_and_Spoke_TLOC_Lists out
 !
!
```

Example 7-7 introduces a new control policy to add Preference values to the TLOCs that are being advertised from WAN Edge routers in the data centers. In Sequence 1, a new matching criterion is configured: **originator**. The originator syntax is **originator** {*originator-ip*}, where *originator-ip* is the system IP address of the WAN Edge device you are looking to match against. In this particular policy example, because the objective is to set different values for WAN Edge devices at the same site ID, matching with site lists would be ineffective. Once the TLOCs have been matched, the desired preference value is specified in the action statement with the syntax **preference** {*value*}. The acceptable range for Preference values is 0 through 4294967295 ($2^{32} – 1$). In Example 7-7, the Preference values 500 and 400 are used. As discussed in Chapter 4, under default conditions, the load is equally shared across available transports. The TLOC Preference attribute permits

Technet24

you to influence traffic outbound and inbound direction preference, with higher values being the most preferred.

You can see the effect of this policy in Figure 7-62, where the output is the result of the IP Route real-time query, which shows only one available next hop, the DC1-Edge1 router.

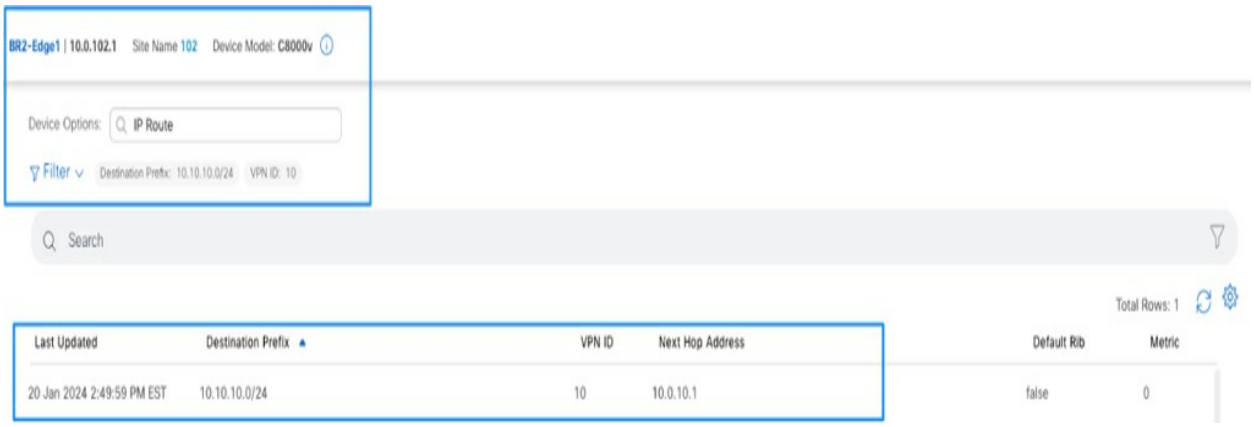


Figure 7-62 *10.0.10.1 [DC1-Edge1] as the Only Next Hop for 10.10.10.0/24*

Figure 7-63 is an updated display of the same output from Figure 7-44 that was taken after the policy in Example 7-7 was applied to the SD-WAN network. In this output, only two paths have been installed in the routing table on BR2-Edge1: the MPLS path and the BIZ-INTERNET path from 10.0.10.1, DC1-Edge1. The output shows all of the routes that have been received from OMP peers in a similar fashion to how the outputs of the Cisco IOS command **show ip bgp** would display all of the routes received from BGP neighbors (even if they are not the best BGP routes or actually installed in the routing table).

Note

In Figure 7-63, in the Search Options field, there is a search for 1.1.1.102, which is the system IP address of SD-WAN Controller-1. This search has the functional effect of removing all the duplicate advertisements that would have been received from SD-WAN Controller-2 (with system IP address 1.1.1.103), making it easier for network administrators to understand the table. Because the same policy should always be applied to all of the SD-WAN Controllers, the same route advertisements should come from both SD-WAN Controllers, and it is safe to filter one set of them from this view.

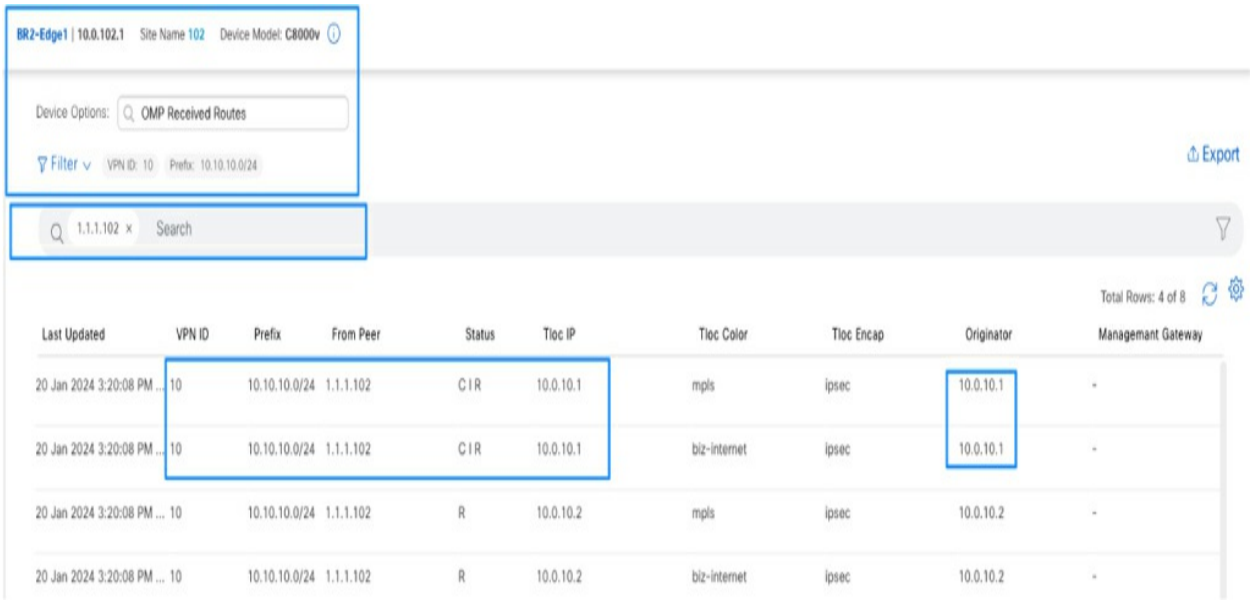


Figure 7-63 *OMP Route Table on BR2-Edge1, Showing Four Paths to 10.10.10.0/24*

Technet24



It is clear from the OMP table that the WAN Edge device is still receiving the OMP route advertisements from DC1-Edge2. These advertisements are the third and fourth entries in the table, with TLOC IP address 10.0.10.2. The Status column for these routes displays R, and the Status column for OMP routes from 10.0.10.1 (the first and second rows) is displayed with the status CIR. The key code can be found in the output of **show sdwan omp routes** from the CLI, as shown in Example 7-8.

Example 7-8 *OMP Status Code Key*

```
BR1-Edge1# show sdwan omp routes
Code:
C -> chosen
I -> installed
Red -> redistributed
Rej -> rejected
L -> looped
R -> resolved
S -> stale
Ext -> extranet
Inv -> invalid
Stg -> staged
```

```
IA  -> On-demand inactive
U   -> TLOC unresolved
BR-R -> Border-Router reoriginated
TGW-R -> Transport-Gateway reoriginated
R-TGW-R -> Reoriginated Transport-Gateway reori
<<<omitted for brevity>>>
```



While the full explanation of these values can be found in the Cisco documentation, a few values are reviewed in Table 7-1.

Table 7-1 *OMP Status Codes*

Status Code	Status Meaning	Explanation
C	Chosen	This route is the successor of the OMP best-path selection process.
I	Installed	The OMP route has been installed into the IP routing table.
R	Resolved	The TLOC referenced in the OMP route is present and operational.

Technet24

The status code C I R means that the OMP route has been successfully installed in the routing table and is being used for forwarding. The third and fourth OMP routes from 10.0.10.2 have the status code R, which means these routes are resolved, but they have not been chosen as the best OMP routes, and they will not be used for forwarding. The reason that these routes have not been chosen can be seen in the output of the OMP TLOCs table, shown in Figure 7-64.

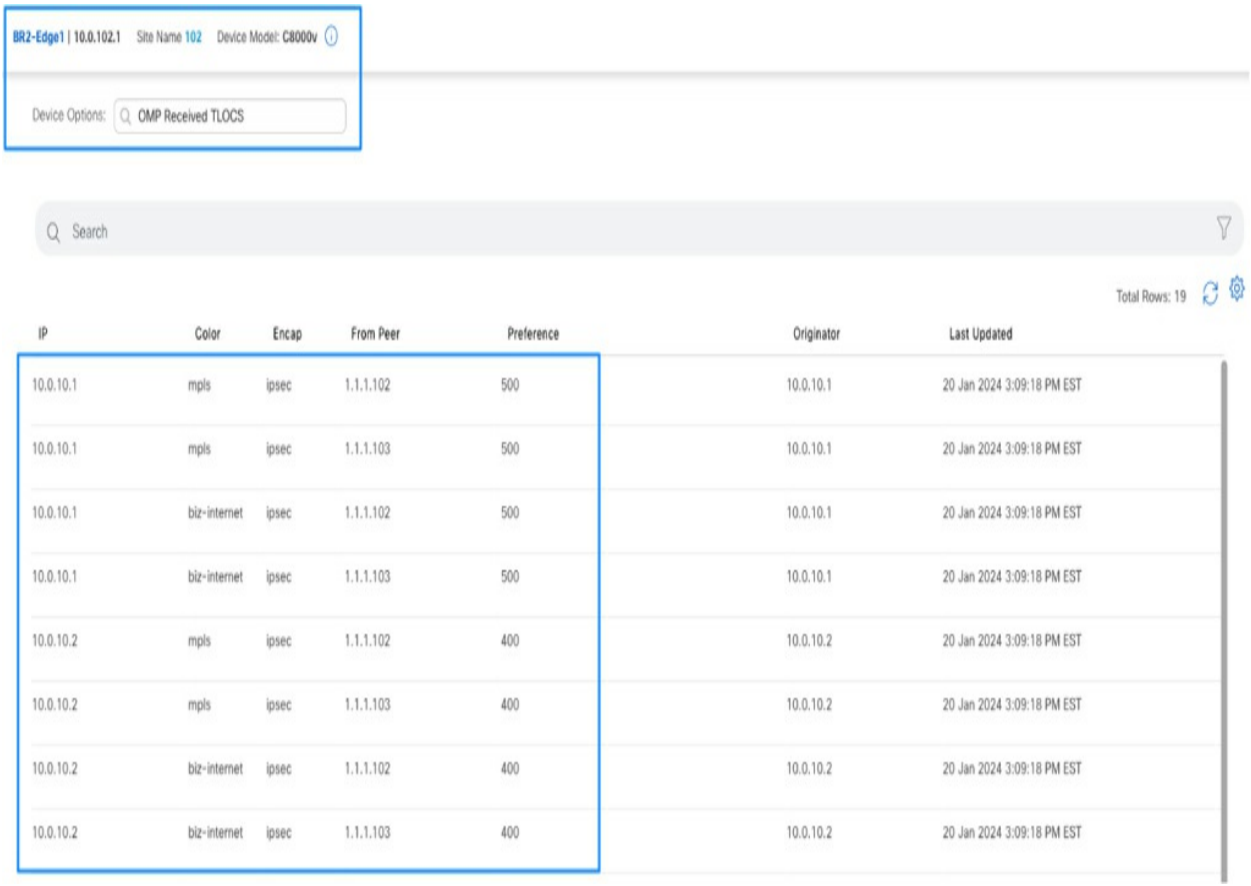


Figure 7-64 *OMP TLOC Table on BR2-Edge2, Showing TLOCs with Different Preference Values*

Since the TLOCs from 10.0.10.1 have a higher Preference value than the

TLOCs from 10.0.10.2, the OMP routes that resolved to the TLOCs from 10.0.10.1 have won the best-path selection process, are chosen, and will subsequently be installed in the routing table.



Figure 7-65 shows the result of this effect, with the TLOCs from 1 and 2 received on WAN Edge 3 with two different TLOC Preference values. As the higher Preference value is received for the TLOCs that are advertised from WAN Edge 1, only the tunnels to WAN Edge 1 will be used for forwarding traffic. If WAN Edge 1 fails, or if both TLOCs are removed from WAN Edge 3, WAN Edge 3 would subsequently use the paths being advertised from WAN Edge 2, and the traffic would fail over to the bottom set of tunnels.

Technet24

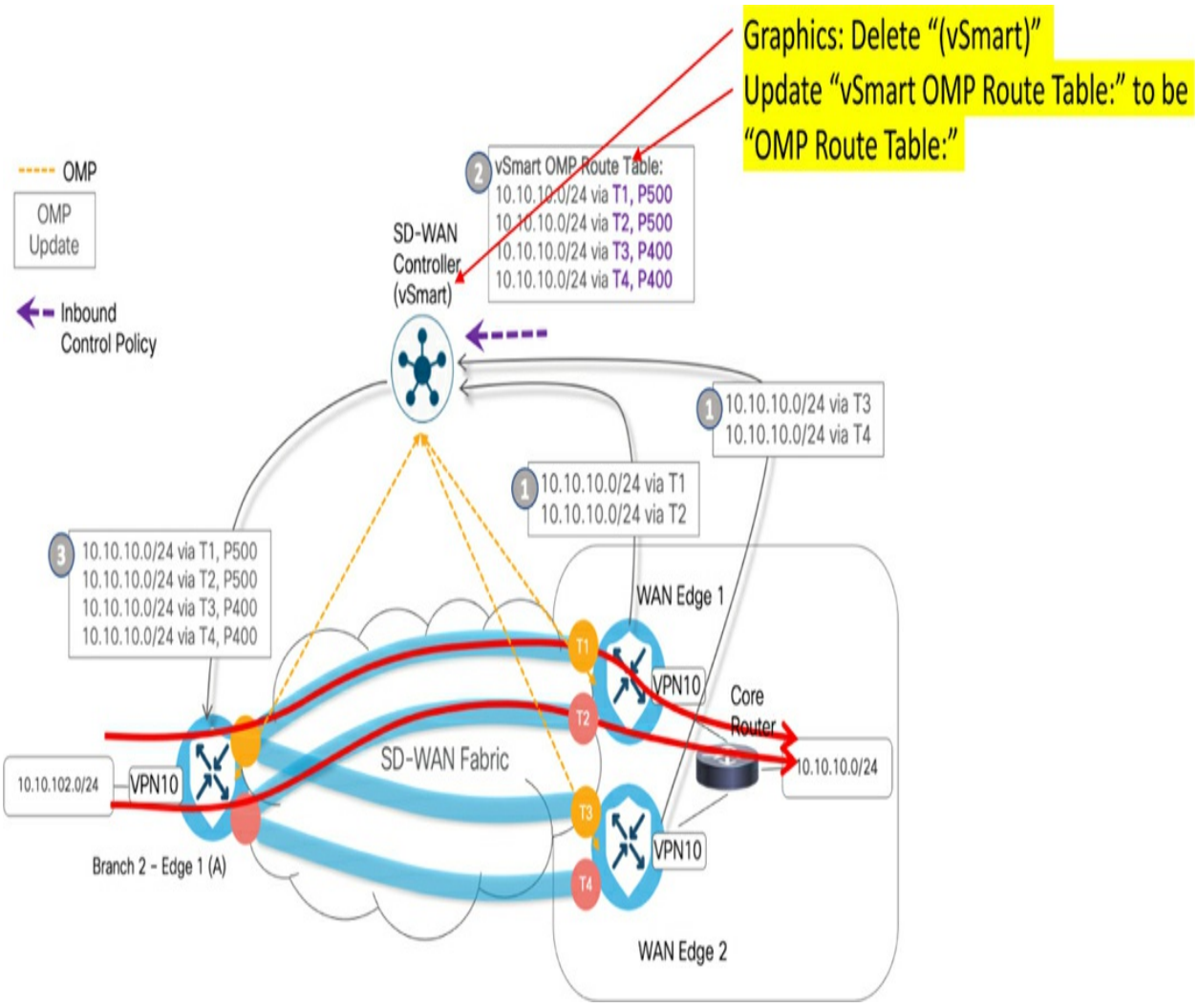


Figure 7-65 *Illustration of the Effects of Different Preference Values on the Forwarding Plane*

Setting TLOC Preference with Device Templates and Configuration Groups

The previous example reviewed the purpose of setting the TLOC Preference value and how to set that value using centralized control policies. In addition to using centralized control policies to set the TLOC Preference value, you can also configure it directly on the tunnel interface. Typically, WAN Edge

router configurations are centrally managed using a combination of feature templates and device templates, as discussed in Chapter 4. While Weight and Preference are configurable on the tunnel interfaces with feature templates, they can also be configured with a CLI add-on template on the tunnel interface directly with the command **encapsulation** {**ipsec**|**gre**} [**preference** *preference*] [**weight** *weight*]. Example 7-9 reviews the CLI configuration used to accomplish these objectives. Starting from SD-WAN Releases 20.9 and 17.9, you can use Configuration Groups to achieve the same purpose by editing the desired TLOC under the Transport and Management profile.

Example 7-9 *Tunnel Interface Configuration with Weight and Preference*

```
! The following configuration excerpts from BR1
! indicate the preference and weight settings t
! on the tunnel interfaces
!
BR1-Edge1# show sdwan running-config sdwan | in
GigabitEthernet|color|encapsulation
 interface GigabitEthernet1
  tunnel-interface
   encapsulation ipsec preference 50 weight 5
   color mpls restrict
 interface GigabitEthernet2
  tunnel-interface
```

Technet24

```
    encapsulation ipsec preference 50 weight 20
    color biz-internet
***Insert 2pt rule
BR1-Edge2# show sdwan running-config sdwan | in
GigabitEthernet|color|encapsulation
 interface GigabitEthernet1
  tunnel-interface
    encapsulation ipsec preference 40 weight 20
    color biz-internet
 interface GigabitEthernet2
  tunnel-interface
    encapsulation ipsec preference 4
    color lte
BR1-Edge2#
```



Example 7-9 shows the Preference attribute, whose values and their purposes we discussed in depth in the previous section. Example 7-9 also includes the use of the Weight attribute, which is used to determine the proportional load-sharing among TLOCs with equal preferences. Weight is not part of the best-path selection algorithm but is instead used after the best paths are determined to forward proportionally among the paths selected through the best-path process. The Weight attribute can range from a value of 1 (default)

to 255. Weight is generally configured in proportion to the bandwidths of the links at a single site. In this example, the Weight settings 20 and 5 could represent links with speeds of 20 and 5 Mbps, respectively. Remember that there is no absolute reference value for Weight; it is only proportional among the transports at the local site.

In Example 7-9, the preferences on the BIZ-INTERNET, MPLS, and LTE links of BR1-Edge1 are configured as 50, 50, and 5, respectively. The preferences on BR1-Edge2 for the BIZ-INTERNET, MPLS, and LTE links are configured as 40, 40, and 4, respectively. As a result of this configuration, in steady-state operation, where all routers and all links are operational, inbound traffic to Branch 1 will be sent over the BIZ-INTERNET and MPLS links of BR1-Edge1. These links are configured with the highest Preference values and have won the best-path selection process. Furthermore, traffic will be load-shared on a per-flow basis of 20:5, or 4:1, based on the configured Weight values. If the MPLS and BIZ-INTERNET links on BR1-Edge1 become inoperable, and the TLOCs are no longer resolvable by other routers in the fabric, the TLOC advertisements from BR1-Edge2's MPLS and BIZ-INTERNET links, configured with a TLOC preference of 40, will win the best-path selection process and will be used by other routers to send traffic to Branch 1. Only if all MPLS and BIZ-INTERNET TLOCs are unresolvable will the LTE TLOC on BR1-Edge1 win the best-path selection process with a preference of 5. Finally, if the LTE TLOC on BR1-Edge1 is unresolvable, then the last TLOC from Branch 1, the LTE TLOC on BR1-Edge2 with a preference of 4, will be selected.

Technet24

You can see the results of these configurations in the TLOCs that have been advertised to DC1-Edge1 in Figure 7-66.

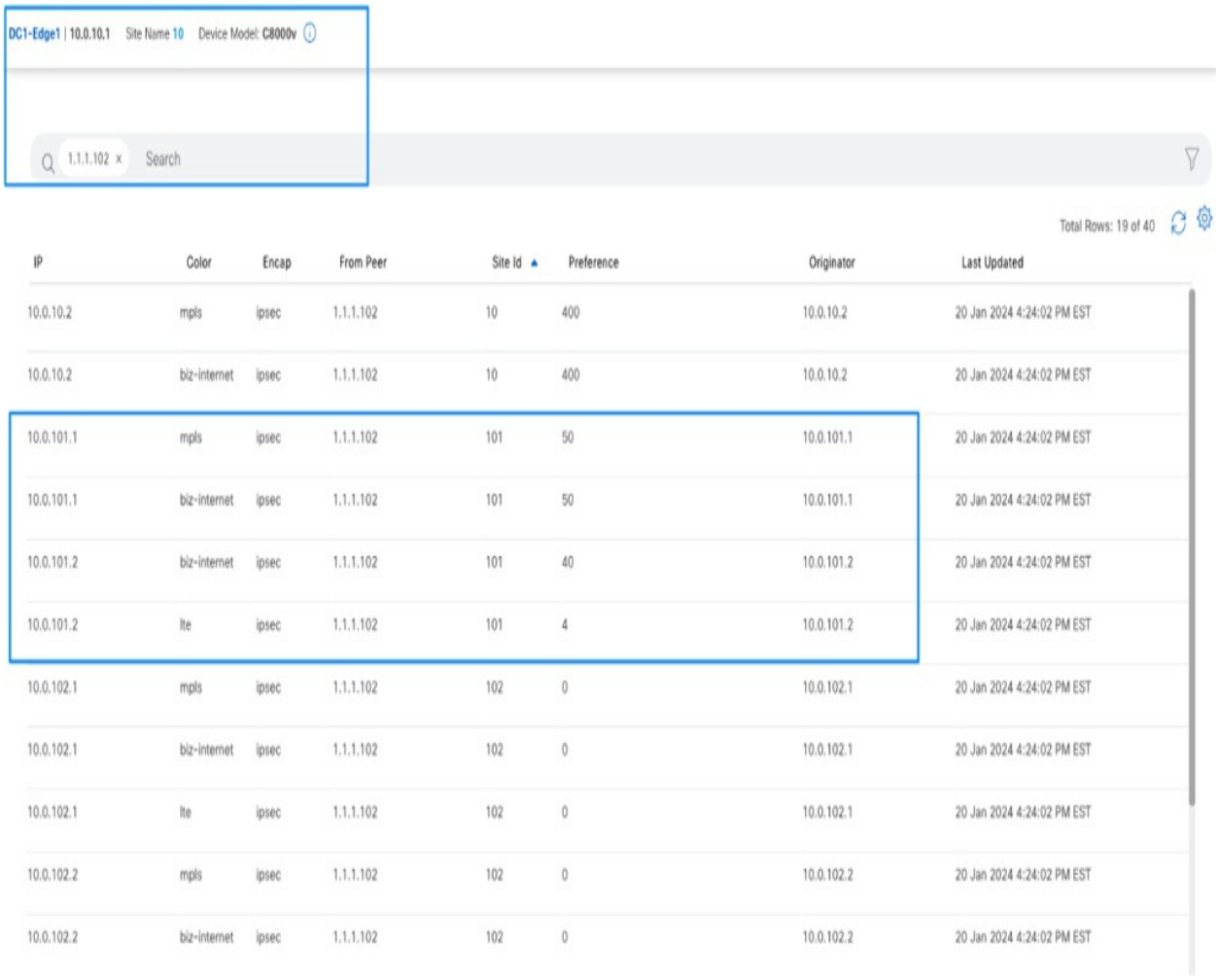


Figure 7-66 *Branch 1 TLOCs Are Received in the DC with the Configured Preference and Weight Values*

Use Case 3 Review

This use case describes two different mechanisms to configure TLOC Preference values and how those Preference values can be used to manipulate

how traffic flows across the network. You also saw how to use the **weight** command to perform unequal-cost load-sharing, which is commonly used when network administrators want to use multiple links of different speeds.

Note

Use Case 3 shows two different ways to set the TLOC Preference values. While the ability to write a centralized control policy to override the preferences that are configured on an interface is useful, the primary use of this policy is for *overriding* the values that were configured on the tunnel interface, not for configuring the values in the first place. We recommend configuring the TLOC Preference values on the WAN Edge tunnel interfaces directly through feature templates or feature parcels and manipulating them with centralized control policies only when necessary.

Note

You may have observed that the choices for Preference values for the data center routers were 500 and 400, while the values used for the Branch 1 location were 50, 40, 5, and 4. While the Preference values have no absolute significance, the decision to use values that were an order of magnitude greater for the data centers than for the branches was by intention. When you

Technet24

configure the data centers with values of 500 and 400 and the branches with lower values, if a prefix (or a default route) that belongs to a data center is ever inadvertently advertised or redistributed from the branch office, the rest of the WAN Edge devices in the fabric will still prefer the advertisements from the data centers because of their higher TLOC preference values. This design choice helps protect the fabric from mistaken or malicious network updates.

Use Case 4: Preferring Regional Data Centers for Internet Access

A common business objective that network administrators need to account for is the desire to have geographically dispersed users access an instance of a shared resource that is close to the users. For example, if a service is accessible from the data centers in New York and London, the users in Boston should generally use the service in New York, whereas the users in Paris and Berlin should generally use the service in London. The definition of a service can be anything—an enterprise ERP application, a video conferencing server, or (as in this use case) a default route to provide Internet access. Figure 7-67 shows the topology for this use case with the configuration objective.

Before we proceed through this use case, it is once more necessary to inject default routes into the VPN 10 routing table from the data centers. As with

Use Case 2, the specific method used to inject the default routes is inconsequential. With the default routes advertised from the data centers and the old policy applied, you can see in Figure 7-68 that there are four default routes installed in the VPN 10 routing table on BR2-Edge1.

Note

The number of paths you can observe for a single prefix at any given time is derived from the best path selection process. This process is influenced by the number of paths WAN Edge routers advertise to SD-WAN Controllers and vice versa. In the current use case, there are more than four routes per prefix, which is also common in Catalyst SD-WAN deployments. OMP configuration settings are available both at the WAN Edge router and the SD-WAN Controller to handle these type of situations. One such setting, **send-backup-paths**, allows the SD-WAN Controller to advertise the first set of non-best routes along with the best routes to WAN Edge devices, which helps to accelerate convergence during brownout situations; this setting directly depends on the total limit path configured at the SD-WAN Controller. Another setting, **send-path-limit**, can be configured on WAN Edge routers to control the maximum number of equal-cost routes sent to SD-WAN Controllers and vice versa. In addition, the **ecmp-limit** setting is useful when it is necessary to control the total number of

Technet24

OMP paths installed in WAN Edge route table. For more details, refer to the Cisco Catalyst SD-WAN Command Reference.

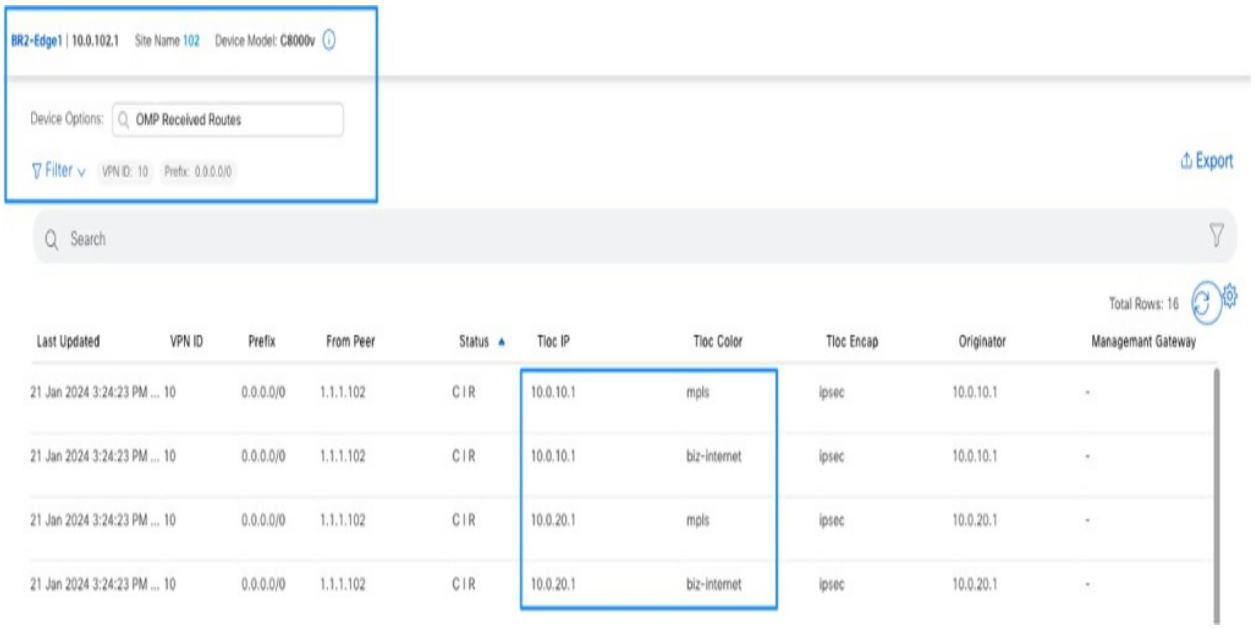


Figure 7-68 *BR2-Edge1 (Paris Branch) Has Installed Four Default Routes in the OMP Routing Table*

The first two of these four routes have the TLOC IP address 10.0.10.1, which is the system IP address of DC1-Edge1. The third and fourth routes have the TLOC IP address 10.0.20.1, which is advertised from DC2-Edge1. This implies that, in this example, the traffic from the Paris branch will egress at the London data center and will be inflicted with the additional latency caused by being routed out the New York data center.

Similarly, for users in the Boston (Branch 1) and Berlin (Branch 3) branches, half of the traffic will egress from the geographically local data center, and

half of the traffic will be sent all the way across the Atlantic Ocean. This is confirmed by the Simulate Flows output from the Boston branch, shown in Figure 7-69.

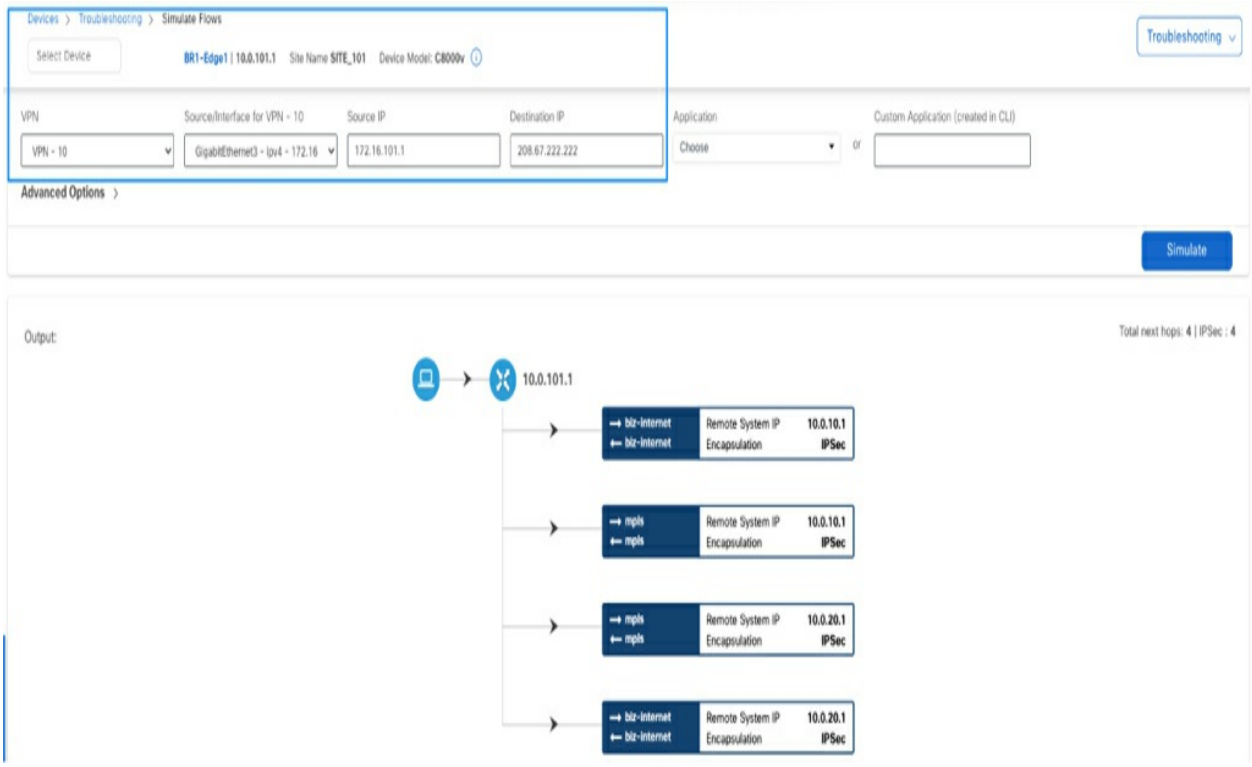


Figure 7-69 *Simulate Flows Output for Destination 208.67.222.222 Confirms That Branch 1 (Boston) Has Four Paths*

Note

The Simulate Flows tool can be found in the Troubleshooting section of Monitor > Devices > {*WAN Edge*}. The Simulate Flows tool can provide the expected forwarding path of a flow given the current state of the WAN Edge device. This output considers the routes in the routing table, the centralized and

Technet24

localized policies that have been applied, and the current performance of the transport links. This can be a very useful tool for seeing if the policies that have been applied will have the intended effect. The Simulate Flows tool does not actually generate and forward any traffic in the data plane; instead, it indicates the path that would be taken if a flow existed to be forwarded.

In order to ensure that users always use the Internet egress closest to their geographic location, you can configure the centralized control policy in Example 7-10 on the network. This policy builds on the policies that were configured for the previous use cases and creates separate outbound control policies such that the European branches will prefer the default routes from the European data centers, and the U.S. branches will prefer the default routes from the U.S. data centers.

Example 7-10 *Use Case 4: Policy for Regionalizing Internet Access*

```
! The control-policy "Hub_and_Spoke_TLOC_Lists"
! Example 7-5 has been changed into two separate
! control policies: Europe_Hub_and_Spoke_TLOC a
North_America_Hub_and_Spoke_TLOC.
policy
```

```
control-policy Europe_Hub_and_Spoke_TLOC
    sequence 1
     match tloc
      site-list DCs
     !
     action accept
     !
    !
! A new sequence was added to this policy to ma
! the London DC (combination of prefix-list and
criteria),
! and set a preference of 100.
!
    sequence 11
     match route
      prefix-list Default_Route
      site-list Europe_DC
     !
     action accept
      set
       preference 100
      !
     !
    !
    sequence 21
```

Technet24

```
      match route
       site-list DCs
       prefix-list _AnyIpv4PrefixList
      !
      action accept
      !
     !
     sequence 31
      match route
       site-list BranchOffices
       prefix-list _AnyIpv4PrefixList
      !
      action accept
       set
        tloc-list DC_TLOCs
       !
      !
     !
   default-action reject
  !
! Similar to the previous control policy, the f
! the default route specifically from the New Y
! sets a preference.  The rest of the policy is
!
 control-policy North_America_Hub_and_Spoke_TLO
```

```
    sequence 1
     match tloc
      site-list DCs
     !
     action accept
     !
    !
    sequence 11
     match route
      prefix-list Default_Route
      site-list North_America_DC
     !
     action accept
      set
       preference 100
      !
     !
    !
    sequence 21
     match route
      site-list DCs
      prefix-list _AnyIpv4PrefixList
     !
     action accept
     !
```

Technet24

```
    !
    sequence 31
     match route
      site-list BranchOffices
      prefix-list _AnyIpv4PrefixList     !
     action accept
      set
       tloc-list DC_TLOCs
      !
     !
    !
 default-action reject
!
control-policy Set_DC_TLOC_Preference
! <<<No changes made to this policy from Exam
!
lists
 ! <<<Some lists without changes from Example 7
 !
 ! A new prefix-list is created to match the d
 !
 prefix-list Default_Route
  ip-prefix 0.0.0.0/0
 !
 site-list BranchOffices
```

```
    site-id 100-199
   !
   site-list DCs
    site-id 10-50
   !
   ! New site lists are created to allow for mor
   ! (Europe_DC and North_America_DC) and more t
scopes
   ! (Europe_Branches and North_America_Branches
   !
   site-list Europe_Branches
    site-id 102-103
   !
   site-list Europe_DC
    site-id 20
   !
   site-list North_America_Branches
    site-id 101
   !
   site-list North_America_DC
    site-id 10
   !
  !
!
! Lastly, the policy that prefers the London DC
```

Technet24

```
! branches, and the policy that prefers the Ame
! American Branches.
!
!
apply-policy
 site-list Europe_Branches
  control-policy Europe_Hub_and_Spoke_TLOC out
 !
 site-list North_America_Branches
  control-policy North_America_Hub_and_Spoke_TL
 !
 site-list DCs
  control-policy Set_DC_TLOC_Preference in
 !
!
```

With this new centralized policy applied to the SD-WAN fabric, the same outputs from the Paris and Boston offices can be evaluated to further understand the policy's effect. In Figure 7-70, you can see that there are now only two routes installed in the routing table on BR2-Edge1 (in contrast to the four routes present before the policy was applied; refer to Figure 7-68).

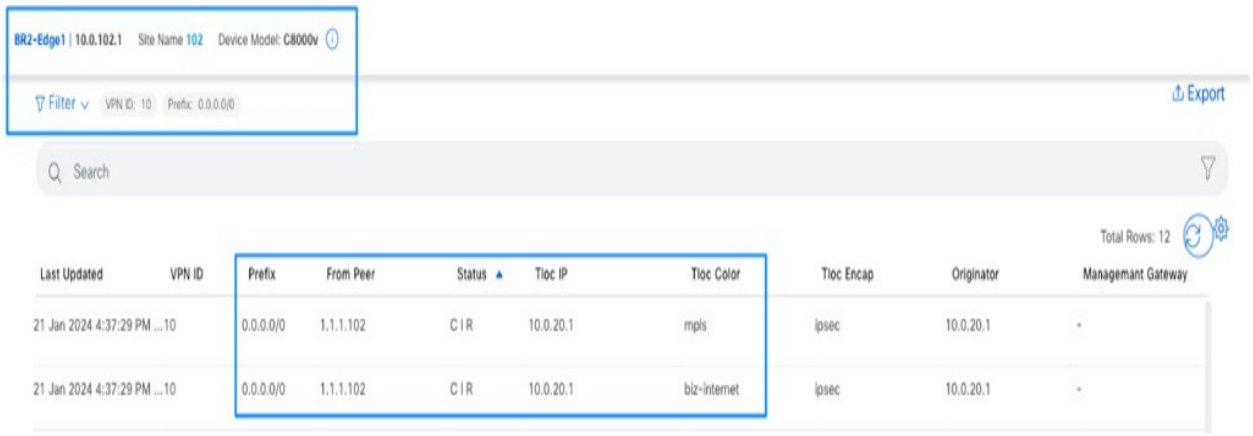


Figure 7-70 *BR2-Edge1 (Paris Branch) Has Two Default Routes Installed in the Routing Table*

The default routes installed on BR2-Edge1 (Paris branch) have the TLOC IP address 10.0.20.1, and they originate from the DC2-Edge1 router (London data center). Further inspection of the OMP Routes table on BR2-Edge1 in Figure 7-71 shows that all eight default routes have been received by the WAN Edge device, but only the two with Route Preference 100 have been selected as the best path and have been inserted into the VPN 10 routing table with the status C I R.

Technet24

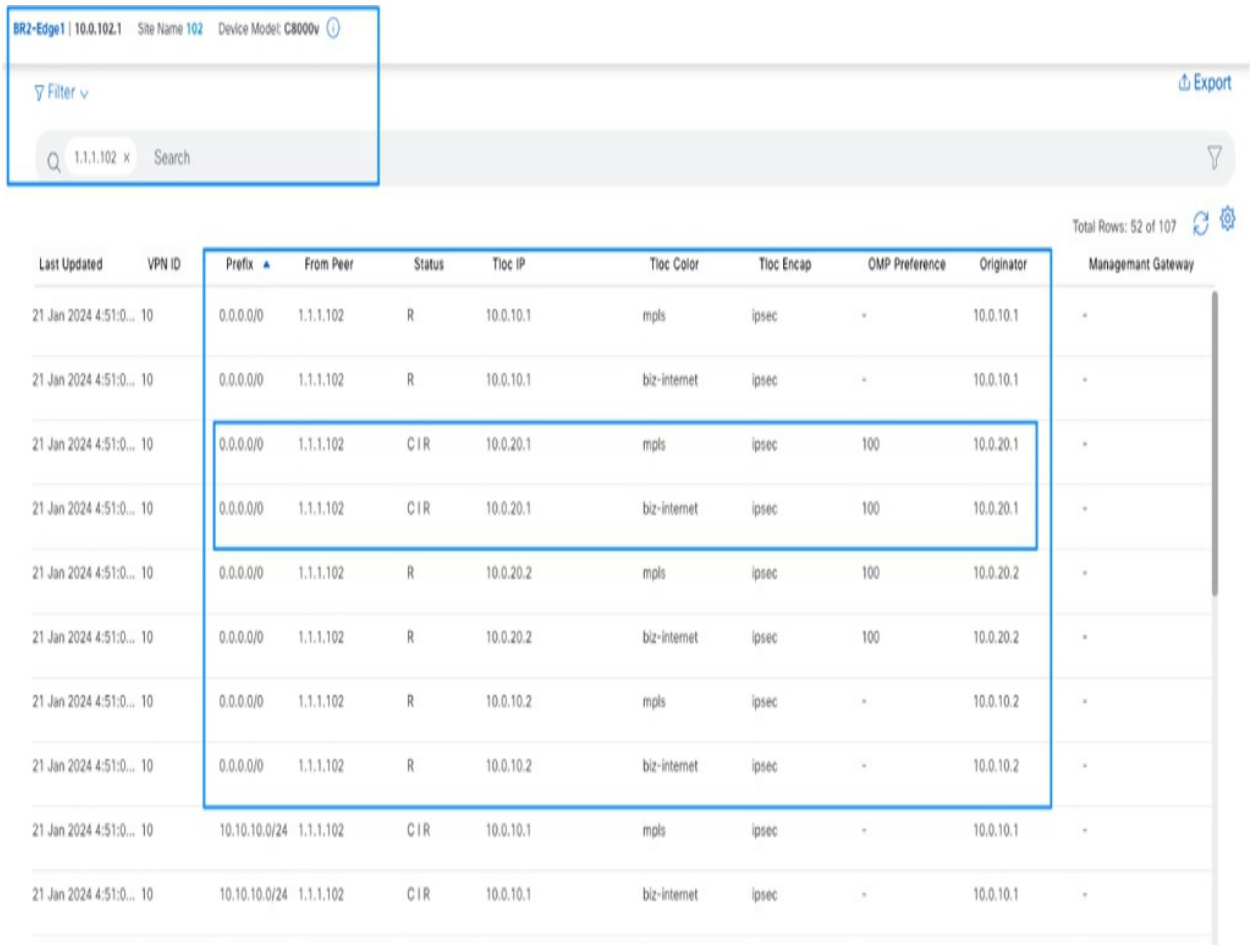


Figure 7-71 *BR2-Edge1 (Paris Branch) Has Installed Eight Default Routes in the OMP Table*

Figure 7-71 shows that BR2-Edge1 has received all eight default routes (four data center WAN Edge routers, each with MPLS and BIZ-INTERNET TLOCs). Of these eight routes, the four that are being advertised from the London data center (site ID 20) have OMP Route Preference 100. Of those four, the two that resolve to the 10.0.20.1 TLOC IP addresses have a TLOC Preference of 500 and will be preferred over the TLOC Preference of 400 for the TLOCs from 10.0.20.2 (from the policy created in Use Case 3). These two preferred routes are being installed into the routing table with of the

status C I R.

The Boston branch has selected two different paths that it will use to forward its traffic toward the Internet, as shown in Figure 7-72.

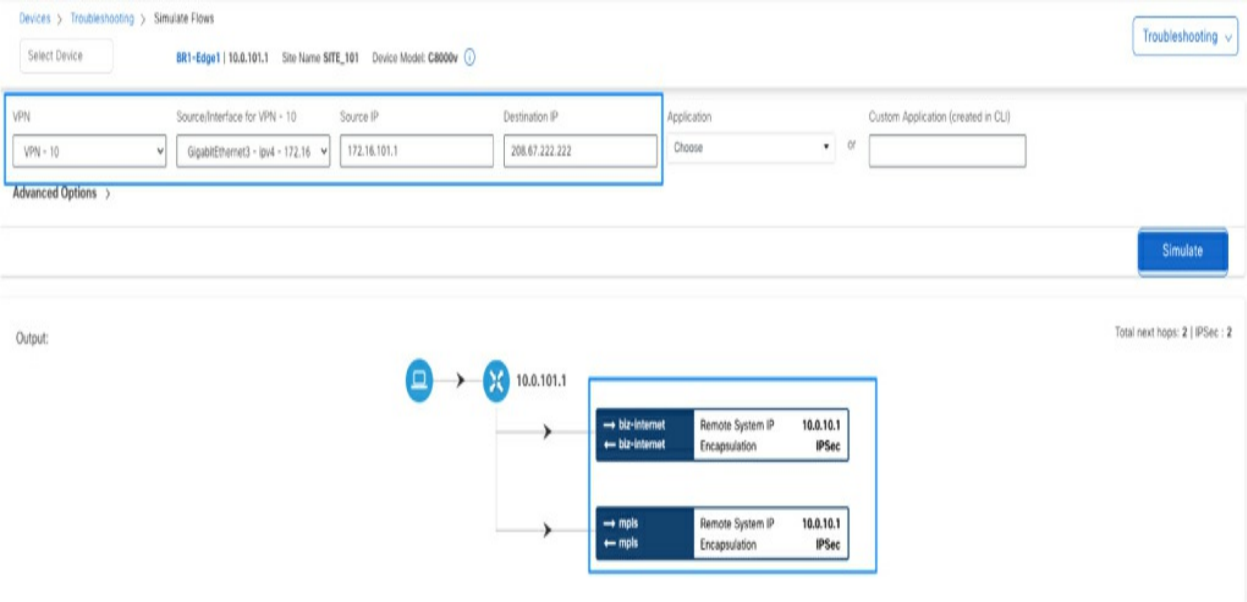


Figure 7-72 *Simulate Flows Output Confirms That Branch 1 (Boston) Will Egress via the New York Data Center (System IP Address 10.0.10.1)*

The Boston branch (BR1-Edge1) will use the two paths from DC1-Edge1 (system IP address 10.0.10.1) that have both a Route Preference of 100 and a TLOC Preference of 500.

Use Case 4 Review

This use case describes how to solve a traffic engineering problem that often plagues network administrators: how to choose the closest instance of a shared resource. While the configurations reviewed in this section may seem

Technet24

overwhelming for a simple network of just five sites, network administrators should consider that the only configuration that would be necessary to expand this design to a network of 500 or 5000 sites would be to update the definitions of the Europe_Branches and North_America_Branches site lists. This ability to have a single, centralized place to manage the WAN fabric is a fundamental part of the power of Cisco SD-WAN.

While the examples in the previous use case focused on TLOC Preference, this use case highlights the OMP Route Preference field, and it demonstrates how these two values can be used together. In the best-path selection algorithm, Route Preference is evaluated before TLOC Preference, and it can therefore be used to override the selection that would have been made based on TLOC Preference values for a specific prefix or group of prefixes. This type of flexibility allows network administrators the extremely fine control necessary to solve the business's traffic engineering objectives.

Use Case 5: Regional Mesh Networks

In Use Case 1, we explored turning the network from the default state of a full-mesh topology into a hub-and-spoke network. While a hub-and-spoke design may meet the needs of some organizations, it may be too rigid for many enterprises. Many times, there are legitimate business purposes for some branch office sites to be in communication with other branch office sites. This communication should be permitted without requiring traversal of a corporate data center. Sometimes organizations implement this type of

network because they have sites that share geographic proximity; other times, organizations choose to implement this type of network based on the business function of a site (such as R&D, manufacturing, sales, and so on). Figure 7-73 shows an existing topology altered to form a regional mesh of the European offices.

Building on the policies that were created in the previous use cases, in this use case we look at building regional mesh networks in order to permit the sites in Europe to communicate directly with other sites in Europe without needing to transit a data center to do so. The new data plane tunnel that will need to be created is highlighted in Figure 7-73. A corresponding policy for North America will also be built, but because there is currently only a single branch site in North America, the policy will not have an effect on the network. Transiting the data centers will still be required in order for a branch office in North America to communicate with a branch office in Europe.

Technet24

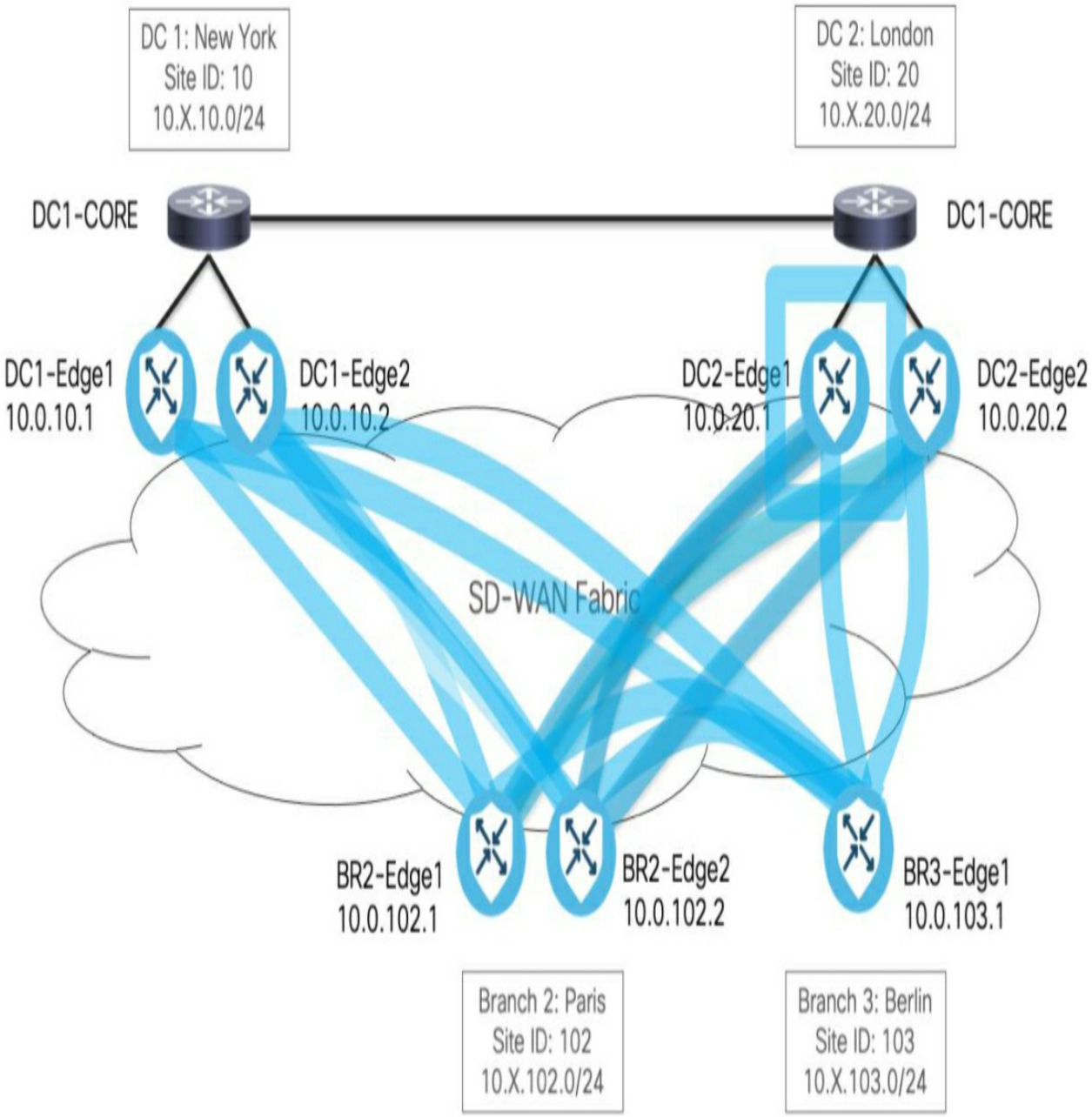
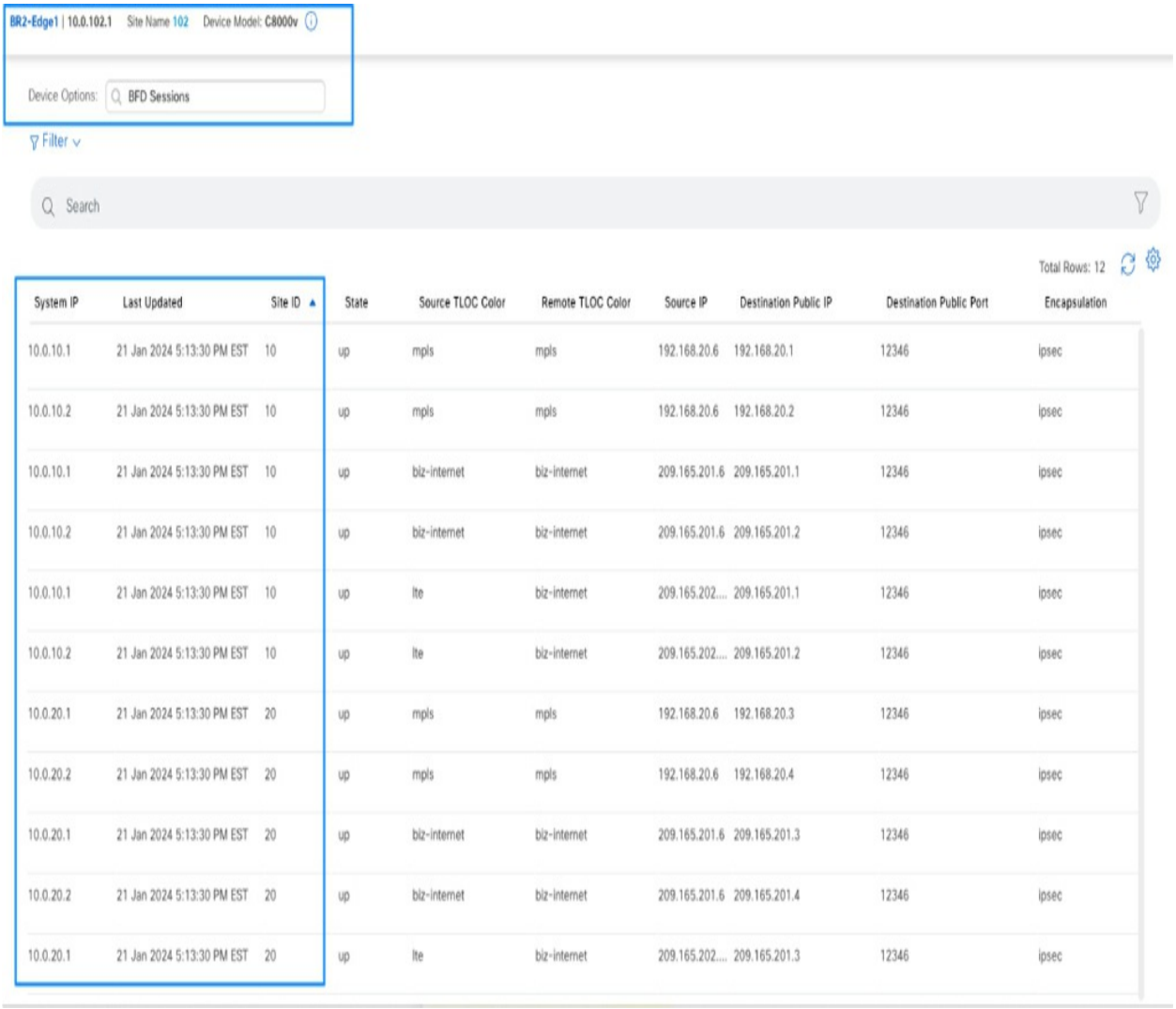


Figure 7-73 *Network Topology with Regional Meshes*

Before the new policy is applied, you can see that there is no direct data plane connection between BR2-Edge1 and BR3-Edge1 when you look at the real-time BFD Sessions output shown in Figure 7-74. All the BFD sessions terminate at data center WAN Edges, as indicated by the system IP addresses

10.0.10.X and 10.0.20.X.



System IP	Last Updated	Site ID	State	Source TLOC Color	Remote TLOC Color	Source IP	Destination Public IP	Destination Public Port	Encapsulation
10.0.10.1	21 Jan 2024 5:13:30 PM EST	10	up	mpls	mpls	192.168.20.6	192.168.20.1	12346	ipsec
10.0.10.2	21 Jan 2024 5:13:30 PM EST	10	up	mpls	mpls	192.168.20.6	192.168.20.2	12346	ipsec
10.0.10.1	21 Jan 2024 5:13:30 PM EST	10	up	biz-internet	biz-internet	209.165.201.6	209.165.201.1	12346	ipsec
10.0.10.2	21 Jan 2024 5:13:30 PM EST	10	up	biz-internet	biz-internet	209.165.201.6	209.165.201.2	12346	ipsec
10.0.10.1	21 Jan 2024 5:13:30 PM EST	10	up	lte	biz-internet	209.165.202....	209.165.201.1	12346	ipsec
10.0.10.2	21 Jan 2024 5:13:30 PM EST	10	up	lte	biz-internet	209.165.202....	209.165.201.2	12346	ipsec
10.0.20.1	21 Jan 2024 5:13:30 PM EST	20	up	mpls	mpls	192.168.20.6	192.168.20.3	12346	ipsec
10.0.20.2	21 Jan 2024 5:13:30 PM EST	20	up	mpls	mpls	192.168.20.6	192.168.20.4	12346	ipsec
10.0.20.1	21 Jan 2024 5:13:30 PM EST	20	up	biz-internet	biz-internet	209.165.201.6	209.165.201.3	12346	ipsec
10.0.20.2	21 Jan 2024 5:13:30 PM EST	20	up	biz-internet	biz-internet	209.165.201.6	209.165.201.4	12346	ipsec
10.0.20.1	21 Jan 2024 5:13:30 PM EST	20	up	lte	biz-internet	209.165.202....	209.165.201.3	12346	ipsec

Figure 7-74 *BR2-Edge1 Does Not Currently Have Any Data Plane Tunnels to 10.0.103.1/Site 103*

The **traceroute** output in Example 7-11 confirms that while Branch 2 has connectivity to Branch 1 and Branch 3, the data plane must currently transit the data center, taking a total of two hops to reach both branch offices.

Example 7-11 *Tracing the Path from 10.10.102.1 to Other Branches via the*

Technet24

Data Center

```
! Tracing a path from BR2 to BR3 is successful
! and transits a datacenter
!
BR2-Edge1# traceroute vrf 10 10.10.103.1
<<<Output omitted for brevity>>>
  1 BIZ-INET-DC1-Edge1 (209.165.201.1) 2 msec 1
  2 MPLS-BR3-Edge1 (192.168.20.8) 1 msec 2 msec
  3 BR3-Edge1-Core (172.16.103.2) 2 msec 2 msec
BR2-Edge1#
!
! Tracing a path from BR2 to BR1 is successful
! and transits a datacenter
!
BR2-Edge1# traceroute vrf 10 10.10.101.1
<<<Output omitted for brevity>>>
  1 BIZ-INET-DC1-Edge1 (209.165.201.1) 1 msec 1
  2 BIZ-INET-BR1-Edge1 (192.168.100.1) 2 msec 2
  3 BR1-Edge1-Core (172.16.101.2) 14 msec 3 mse
BR2-Edge1#
```

In order to enable a regional mesh for the European locations, the policy in

Example 7-12 is applied. In order to form regional meshes, the SD-WAN Controllers will need to advertise the TLOC routes and the routes for the other sites within the mesh with which connectivity should be established. For sites that are outside the regional mesh, the TLOCs will not be advertised, and the OMP routes will be advertised with the data center TLOCs.

Example 7-12 *Use Case 5: Policy for Establishing Regional Mesh Data Planes*

```
! In the "To_Europe_Branches_Regional_Mesh" co
sequence 41 were
! added to permit the advertisements of the TL
from
! other sites in the Site List "Europe_Branche
has
! been updated so that only the sites in the "
list
! now have their TLOCs updated with the TLOC l
!

policy
 control-policy To_Europe_Branches_Regional_Me
    sequence 1
```

Technet24

```
      match tloc
       site-list DCs

      !
      action accept
      !
     !
     sequence 11
      match tloc
       site-list Europe_Branches
      !
      action accept
      !
     !
     sequence 21
      match route
       prefix-list Default_Route
       site-list Europe_DC
      !
      action accept
       set
        preference 100
       !
      !
     !
```

```
    sequence 31
     match route
      site-list DCs
      prefix-list _AnyIpv4PrefixList
     !
     action accept
     !
    !
    sequence 41
     match route
      site-list Europe_Branches
      prefix-list _AnyIpv4PrefixList
     !
     action accept
     !
    !
    sequence 51
     match route
      site-list North_America_Branches
      prefix-list _AnyIpv4PrefixList
     !
     action accept
      set
       tloc-list DC_TLOCs
      !
```

Technet24

```
     !
    !
  default-action reject
 !
! Similar to the previous control policy, the t
TLOCs
! and Routes from the "North_America_Branches"
! Additionally, sequence 51 has been updated to
! Branches.
!
control-policy To_North_America_Branches_Regior
    sequence 1
     match tloc
      site-list DCs
     !
     action accept
     !
    !
    sequence 11
     match tloc
      site-list North_America_Branches
     !
     action accept
     !
    !
```

```
    sequence 21
     match route
      prefix-list Default_Route
      site-list North_America_DC
     !
     action accept
      set
       preference 100
      !
     !
    !
    sequence 31
     match route
      site-list DCs
      prefix-list _AnyIpv4PrefixList
!     action accept
     !
    !
    sequence 41
     match route
      site-list North_America_Branches
      prefix-list _AnyIpv4PrefixList
     !
     action accept
     !
```

Technet24

```
    !
    sequence 51
     match route
      site-list Europe_Branches
      prefix-list _AnyIpv4PrefixList
     !
     action accept
      set
       tloc-list DC_TLOCs
      !
     !
    !
  default-action reject
 !
 control-policy Set_DC_TLOC_Preference
 ! <<<No changes made to this policy from Exam
 !
 lists
 ! <<<No changes made to the lists from Example
 !
!
! Lastly, the new policies are applied to the s
!
apply-policy
 site-list Europe_Branches
```

```
 control-policy To_Europe_Branches_Regional _N
!
site-list North_America_Branches
 control-policy To_North_America_Branches_Regi
!
site-list DCs
 control-policy Set_DC_TLOC_Preference in
!
```

After the policy in Example 7-12 is applied to the network, the necessary TLOC routes are exchanged between BR2-Edge1 (Paris) and BR3-Edge1 (Berlin) in order to form data plane tunnels. You can see these five additional tunnels between the two sites in the BFD Sessions output from BR2-Edge1 (see Figure 7-75).

Technet24

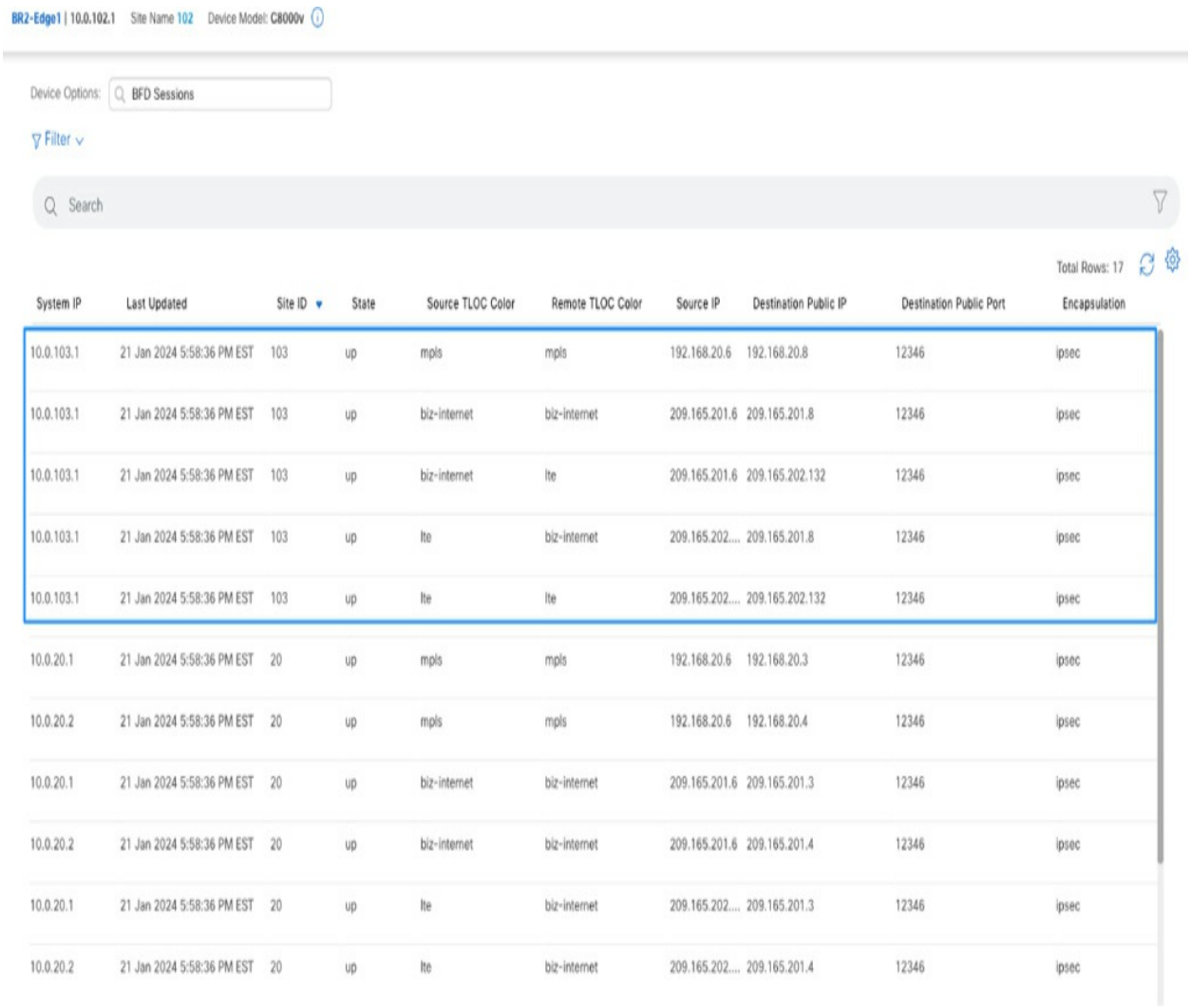


Figure 7-75 *BR2-Edge Has New Data Plane Tunnels to 10.0.103.1/Site 103*

The data plane connectivity can be validated by testing with **traceroute**, as shown in Example 7-13. The first trace shows that there is now a direct path established between Paris and Berlin as part of the European regional mesh. There is no longer a need to transit a data center router to facilitate that connectivity, and the path is now only a single hop. However, as the second **traceroute** output indicates, there is no direct data plane path from Paris to Boston. Traffic that is going from Branch 2 (Paris) or Branch 3 (Berlin) to

Branch 1 (Boston) must still transit the data center.

Example 7-13 *Tracing the Path from 10.10.102.1 to Other Branches After the Regional Mesh Is Established*

```
! Tracing a path from BR2 to BR3 is successful
!
BR2-Edge1# traceroute vrf 10 10.10.103.1
<<<Output omitted for brevity>>>
  1 BIZ-INET-BR3-Edge1 (209.165.201.8) 2 msec 1
  2 BR3-Edge1-Core (172.16.103.2) 1 msec 2 msec
BR2-Edge1#
!
! Tracing a path from BR2 to BR1 is successful,
indirect and transits a datacenter
!
BR2-Edge1# traceroute vrf 10 10.10.101.1
<<<Output omitted for brevity>>>
  1 BIZ-INET-DC1-Edge1 (209.165.201.1) 1 msec 1
  2 BIZ-INET-BR1-Edge1 (192.168.100.1) 1 msec 1
  3 BR1-Edge1-Core (172.16.101.2) 2 msec 2 msec
BR2-Edge1#
```

Use Case 5 Review

Technet24

This use case shows how to alter the structure of the SD-WAN fabric so that it is not purely a hub-and-spoke network and instead consists of multiple distinct regional mesh networks. Network administrators may deploy regional meshes to permit offices that are in close proximity to each other or offices that serve the same business function to communicate without the need for a data center to proxy the communication.

This use case also continues to highlight how trivial it is to create increasingly complex topologies through a single, centrally administered control policy. The only thing necessary to turn the hub-and-spoke network into a regional mesh network was to advertise the TLOC routes and OMP routes to the sites that need to be able to communicate with each other.

Use Case 6: Enforcing Security Perimeters with Service Insertion

Enterprises often have a need to inject services into the traffic path of the WAN environment. One of the most frequently used services is a security device such as a firewall. Other potential services include IPSs/IDSs, network sniffers, web proxies, caching engines, and WAN optimization appliances. Regardless of the specific type of service, Cisco SD-WAN can natively advertise the availability of the service across the WAN environment and direct traffic flows to that service.

In this use case, there is a need to ensure that users at branch sites in Europe cannot communicate directly with branch sites in North America without first

having their traffic inspected by a firewall and vice versa. We are going to continue to build on the policies created for the previous use cases and modify them to meet this objective.

The first step in service insertion is the configuration of the service itself. For this use case, we are going to provision a firewall attached to DC1-Edge1 as a network service, as shown in Figure 7-76. Traffic flows that originate at a European branch site, such as the example in the diagram, must be inspected by this firewall before being forwarded to a branch location in North America.

Technet24

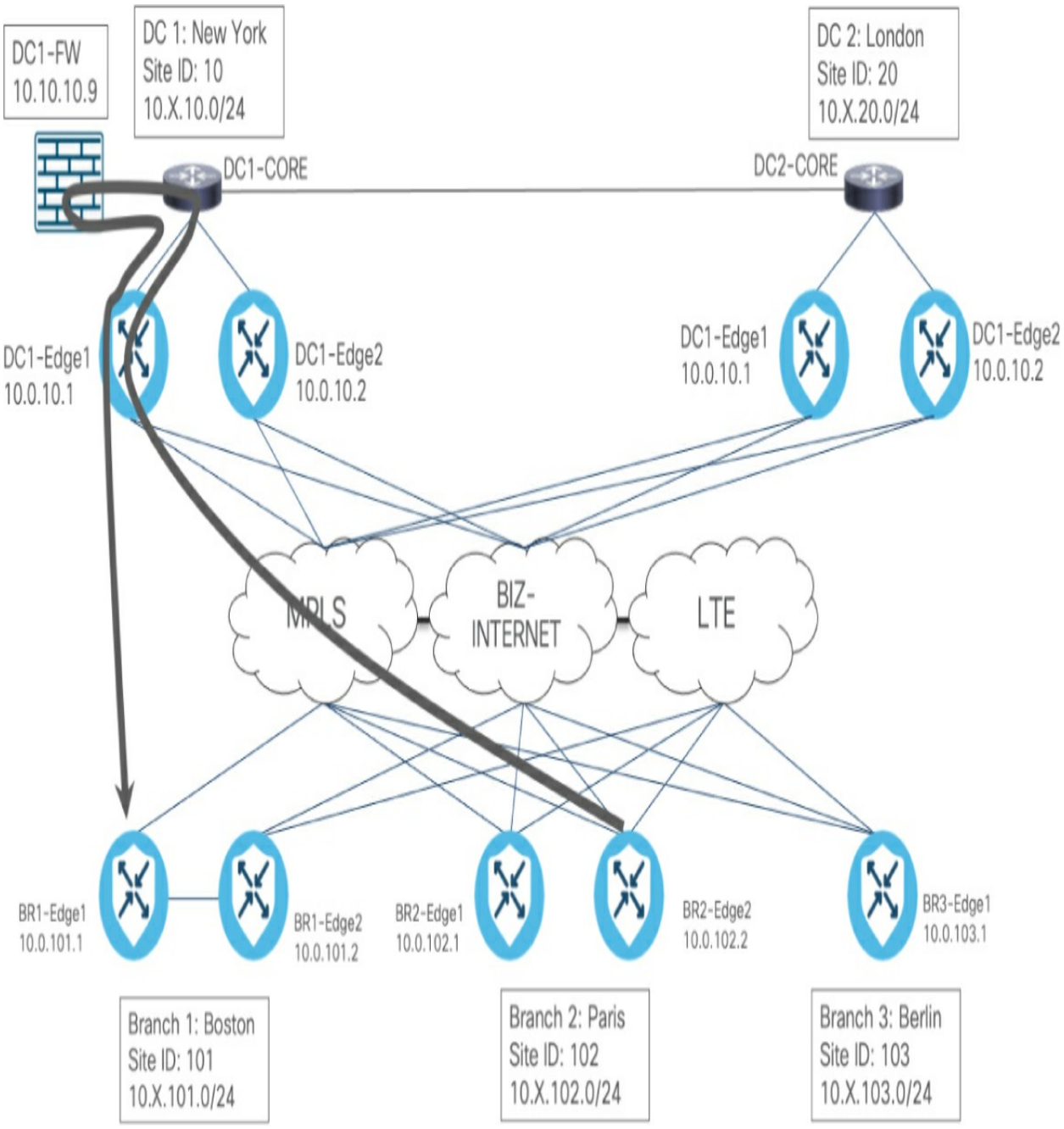


Figure 7-76 *Using a Firewall at the Data Center to Inspect Flows Transiting Between Two Branch Sites*

Example 7-14 shows the configuration needed to create the service in DC1-Edge1 and the advertisements on the SD-WAN Controller as soon as the service is provisioned on the WAN Edge device. Once the service has been

provisioned, a new label is assigned to indicate which traffic received by this WAN Edge device should be forwarded to the service. As Example 7-14 shows, the label in this case is 1004.

Example 7-14 *Firewall Service Configuration on DC1-Edge1*

```
! The minimum configuration necessary for a ser
! specifies where the service is reachable.
!
DC1-Edge1# show sdwan running-config sdwan
sdwan
 service firewall vrf 10
  no track-enable
  ipv4 address 10.10.10.9
!
! Output omitted for brevity
!
! As soon as the service is configured, it is a
Controller and
! ready to be reflected to the entire fabric. C
! send traffic to this service should use label
!
DC1-Edge1# show sdwan omp services
ADDRESS
```

Technet24

```
GROUP
FAMILY   TENANT    VPN    SERVICE  ORIGINATOR
NUMBER      LABEL    STATUS    VRF
-----------------------------------------------
-----------------------------------------
ipv4     0         10     FW      10.0.10.1   0
None        1004    C,Red,R   10

68       None         None           1004
```

A locally attached service, such as the firewall in this example, is configured with the syntax **service** {*service-name*} **vrf** {*vrf-number*} **ipv4 address** {*ip address*}.

Before the policy is applied to the SD-WAN fabric, Branch 2 is able to reach Branch 1 in two hops, as indicated in Example 7-15. Furthermore, this traffic is being forwarded with the label 1003, which represents VPN 10.

Example 7-15 *Tracing the Path from 10.10.102.1 to Branch 1 Before Service Insertion Is Activated*

```
! Tracing a path from BR2 to BR1 is successful
! a datacenter.
!
```

```
BR2-Edge1# traceroute vrf 10 10.10.101.100
Type escape sequence to abort.
Tracing the route to 10.10.101.100
VRF info: (vrf in name/id, vrf out name/id)
  1 MPLS-DC1-Edge1 (192.168.20.1) 2 msec 1 msec
  2 BIZ-INET-BR1-Edge1 (192.168.100.1) 2 msec 1
  3 BR1-Edge1-Core (172.16.101.2) 1 msec 2 msec
  4 BR1-PC1 (10.10.101.100) 1 msec 3 msec 2 msec
BR2-Edge1#
!
! The OMP Route table indicates that the label
! 10.10.101.0/24 without the service insertion
!
BR2-Edge1# show sdwan omp routes 10.10.101.0/24

-----------------------------------------------
omp route entries for tenant-id 0 vpn 1 route 1
-----------------------------------------------
            RECEIVED FROM:
peer            1.1.1.102
path-id         3
label               1003
status              C,I,R
loss-reason     not set
lost-to-peer    not set
```

Technet24

```
lost-to-path-id not set
    Attributes:
     originator       10.0.101.1
     type             installed
     tloc             10.0.10.1, mpls, ipsec
<<Output omitted for brevity>>>
            RECEIVED FROM:
peer            1.1.1.102
path-id         67108867
label               1003
status              C,I,R
loss-reason     not set
lost-to-peer    not set
lost-to-path-id not set
    Attributes:
     originator       10.0.101.1
     type             installed
     tloc             10.0.10.1, biz-internet,
<<Output omitted for brevity>>>
```

Once the firewall service is configured on the WAN Edge router, only a single line of the centralized control policy needs to be changed in order to direct the branch sites to use the service, as shown in Example 7-16.

Example 7-16 *Use Case 6: Service Insertion for Firewalling Branch-to-Branch Transatlantic Traffic*

```
! The only change that has been made is to char
! 51 from referencing the TLOC Lists to now ref
!
policy
 control-policy To_Europe_Branches_Reg_Mesh_wit
    sequence 1
     match tloc
      site-list DCs
     !
     action accept
     !
    !
   sequence 11
     match tloc
      site-list Europe_Branches
     !
     action accept
     !
    !
    sequence 21
     match route
```

Technet24

```
      prefix-list Default_Route
      site-list Europe_DC
     !
     action accept
      set
       preference 100
      !
     !
    !
    sequence 31
     match route
      site-list DCs
      prefix-list _AnyIpv4PrefixList
     !
     action accept
    !
    !
    sequence 41
     match route
      site-list Europe_Branches
      prefix-list _AnyIpv4PrefixList
     !
     action accept
     !
    !
```

```
    sequence 51
     match route
      site-list North_America_Branches
      prefix-list _AnyIpv4PrefixList
     !
     action accept
      set
      service  FW
      !
     !
    !
  default-action reject
 !
 control-policy To_North_America_Branches_Reg_M
  ! The change in this policy mirrors the chang
  ! To_Europe_Branches_Reg_Mesh_with_FW policy
 !
 control-policy Set_DC_TLOC_Preference
  ! No changes made to this policy from Example
 !
 lists
 ! No changes made to the lists from Example 7-1
 !
 !
!
```

Technet24

```
 apply-policy
  site-list Europe_Branches
   control-policy To_Europe_Branches_Reg_Mesh_wi
  !
  site-list North_America_Branches
   control-policy To_North_America_Branches_Reg_
  !
  site-list DCs
   control-policy Set_DC_TLOC_Preference in
  !
 !
```

Once the service insertion policy has been activated, the OMP route table is updated to reflect the new label that will be used to forward traffic from BR2-Edge1 to 10.10.101.0/24. Note that the new label in Example 7-17 is 1004, which is the same label associated with the firewall service in Example 7-14.

The additional hops in the path from Branch 2 to Branch 1 are related to the firewall that the policy requires these transatlantic flows to transit. At the same time, flows between Branch 2 and Branch 3, both part of the European mesh, go directly between the WAN Edge devices and are not required to be filtered through the firewall.

Example 7-17 *Tracing the Path from 10.10.102.1 to Branch 1 After Service*

Insertion Is Activated

```
! The OMP Route table indicates that the label
! 10.10.101.0/24 with the service insertion po
!
BR2-Edge1# show sdwan omp routes 10.10.101.0/24
-----------------------------------------------
omp route entries for tenant-id 0 vpn 10 route
-----------------------------------------------
            RECEIVED FROM:
peer            1.1.1.102
path-id         268435459
label               1004
status          C,I,R
loss-reason     not set
lost-to-peer    not set
lost-to-path-id not set
    Attributes:
     originator       10.0.101.1
     type             installed
     tloc             10.0.10.1, mpls, ipsec
<<omitted for brevity>>>
            RECEIVED FROM:
peer            1.1.1.102
```

Technet24

```
path-id         335544323
label               1004
status          C,I,R
loss-reason     not set
lost-to-peer    not set
lost-to-path-id not set
    Attributes:
     originator       10.0.101.1
     type             installed
     tloc             10.0.10.1, biz-internet,
<<omitted for brevity>>>
!
! Tracing a path from BR2 to BR1 is successful,
! has additional hops for the Firewall Service
!
BR2-Edge1# traceroute vrf 10 10.10.101.100
<<<Output omitted for brevity>>>
  1 MPLS-DC1-Edge1 (192.168.20.1) 2 msec 1 msec
  2 DC1-Edge1-Core (172.16.10.2) 2 msec 1 msec
  3 DC1-FW-SVC (10.10.10.9) 2 msec 1 msec 1 mse
  4 DC1-Edge1-Core-Self (172.16.10.1) 2 msec 1
  5 BIZ-INET-BR1-Edge1 (192.168.100.1) 3 msec 3
 6 BR1-Edge1-Core (172.16.101.2) 1 msec 2 msec
7 BR1-PC1 (10.10.101.100) 1 msec 3 msec 2 msec
```

```
BR2-Edge1# traceroute vrf 10 10.10.103.1
Type escape sequence to abort.
Tracing the route to 10.10.103.1
VRF info: (vrf in name/id, vrf out name/id)
  1 BIZ-INET-BR3-Edge1 (209.165.201.8) 1 msec 1
  2 BR3-Edge1-Core (172.16.103.2) 2 msec 1 msec
BR2-Edge1#
```

Use Case 6 Review

This use case reviews how to provision a network service and manipulate the control policy so that traffic flows are directed to that service. This example uses a single firewall that is advertised throughout the fabric. In the real world, it is much more common to see redundant sets of network services distributed throughout the WAN fabric. In those cases, the lessons from the previous use cases can be combined with network service insertion to ensure that users have reliable, fast access to the services they need and that the services fail over in a predictable manner.

The traffic engineering completed in this use case is all based on the control plane; the policy is keyed off a specific set of source and destination sites and prefixes. In Chapter 8, "Centralized Data Policies," we will explore how to continue to enhance this policy so that specific applications can be directed

Technet24

through network services rather than through just an entire site or subnet.

Use Case 7: Isolating Guest Users from the Corporate WAN

Many organizations have sites where guests are permitted onto the network. Basic security can be provided by confining guest users to their own VPN. However, the default behavior of the SD-WAN fabric is such that connectivity within a single VPN is automatically established between different sites through the exchange of TLOC routes and OMP routes. While this automatic behavior is desired for the corporate network segments, most organizations do not want to permit guest users to communicate with other guest users across the WAN fabric. In this use case, we will explore the use of VPN membership policies to prohibit the exchange of control plane information from VPN 50 (the guest user VPN) to the SD-WAN Controllers.

Before any policy is applied to the network, users in the guest VPN (VPN 50) are able to reach other guest users at different sites, as shown in Example 7-18.

Example 7-18 *Guest VPN Connectivity Between Different Sites*

```
! BR2-Edge1 is able to reach BR3-Edge1 in the g
!
BR2-Edge1# ping vrf 50 10.50.103.1
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 10.50.103.1,
!!!!!
```

In order to prohibit this type of site-to-site connectivity in the guest VPN, we can add a VPN membership policy to the centralized control policy. Example 7-19 shows the centralized control policy to implement this change.

Example 7-19 *Use Case 7: VPN Membership Policy to Prohibit Guest VPN Connectivity Between Different Sites*

```
! The new piece of this policy is the VPN Membe
membership
! policy permits the VPNs for Corporate (VPN 1)
! VPNs will be subject to the default action (r
policy
 vpn-membership vpnMembership_-950781881
    sequence 10
     match
      vpn-list CorporateVPN
     !
     action accept
     !
    !
```

Technet24

```
    sequence 20
     match
      vpn-list PCI_VPN
     !
     action accept
     !
    !
  default-action reject
!
 control-policy Set_DC_TLOC_Preference
  ! <<<No changes made to this policy from Exar
brevity>>>
 !
 control-policy To_Europe_Branches_Reg_Mesh_wit
  ! <<<No changes made to this policy from Exar
brevity>>>
 !
 control-policy To_North_America_Branches_Reg_M
  ! <<<No changes made to this policy from Exar
brevity>>>
 !
 lists
  ! <<<Some lists without changes from Example 7
  !
  ! Two new VPN lists were created to work with
```

```
   !
   vpn-list CorporateVPN
    vpn 10
   !
   vpn-list PCI_VPN
    vpn 20
   !
  !
!
! Lastly, the VPN Membership policy is applied
!
apply-policy
 site-list Europe_Branches
  control-policy To_Europe_Branches_Reg_Mesh_wi
 !
 site-list North_America_Branches
  control-policy To_North_America_Branches_Reg_
 !
 site-list DCs
  control-policy Set_DC_TLOC_Preference in
 !
 site-list BranchOffices
  vpn-membership vpnMembership_-950781881
 !
!
```

Technet24



As Example 7-19 shows, VPN membership policies follow a similar structure and syntax to the other centralized control policies that we have been working with throughout this chapter. In the most basic sense, a VPN membership policy specifies which VPNs will be permitted to join the fabric from a specific site. The remaining VPNs are rejected by the default action.

You can clearly see the effect of the VPN membership policy by looking at the OMP services output on the SD-WAN Controller, as shown in Example 7-20. The command **show omp services** for VPN 3 lists the service as rejected. In this state, the SD-WAN Controller will not propagate any incoming updates from VPN 3 to other OMP peers, and the SD-WAN Controller will not forward updates about VPN 3 from other WAN Edge routers to these locations that have been rejected. Functionally, this isolates VPN 3 from a control plane perspective. This is proven out in Example 7-20 when users in the guest segment at Branch 2 are no longer able to reach the guest segments at other locations.

Example 7-20 *Use Case 7: Effects of the VPN Membership Policy*

```
vSmart-1# show omp services family ipv4 vpn 50
C   -> chosen
```

```
I   -> installed
Red -> redistributed
Rej -> rejected
L   -> looped
R   -> resolved
S   -> stale
Ext -> extranet
Inv -> invalid
Stg -> staged
IA  -> On-demand inactive
U   -> TLOC unresolved

                                                      PA
VPN    SERVICE  ORIGINATOR  FROM PEER         ID
-------------------------------------------------------
50     VPN      10.0.102.1  10.0.102.1        66
                            10.0.102.1        68
                            10.0.102.1        70
50     VPN      10.0.102.2  10.0.102.2        66
                            10.0.102.2        68
                            10.0.102.2        70
50     VPN      10.0.103.1  10.0.103.1        66
                            10.0.103.1        68
                            10.0.103.1        70
!
```

Technet24

```
!
! BR2-Edge1 is unable to reach BR3-Edge1 in the
!
BR2-Edge1# ping vrf 50 10.50.103.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.50.103.1,
U.U.U
Success rate is 0 percent (0/5)
BR2-Edge1#
!
! Guest Users at BR2-Edge1 are still able to ac
! local internet egress even though they can no
sites.
!
BR2-Edge1# ping vrf 50  208.67.222.222
Sending 5, 100-byte ICMP Echos to 208.67.222.22
!!!!!
Success rate is 100 percent (5/5), round-trip r
BR2-Edge1#
```

Use Case 7 Review

This use case shows how to use a VPN membership policy to prohibit the guest VPN from exchanging control plane information with the SD-WAN

Controllers and in turn prohibit the guest VPN from forwarding traffic across the SD-WAN fabric. While a VPN membership policy can be used to prohibit a VPN from being used at all over the SD-WAN fabric, it can also be used to protect and secure VPNs by prohibiting a WAN Edge device that is provisioned either mistakenly or maliciously with a sensitive VPN from being able to exchange control plane information and access the data plane.

Use Case 8: Creating Different Network Topologies for Each Segment

In this use case, we will continue to revise and combine some of the earlier policies that we have discussed and apply them to different network segments at the same time. Currently, there is a European regional mesh that connects Branch 2 and Branch 3. In this use case, we will keep that forwarding path available for the corporate users in VPN 10, but we will revert the PCI VPN (VPN 20) back to a hub-and-spoke topology.

Before the policy is activated, the output in Example 7-21 confirms that currently both VPN 10 and VPN 20 have direct data plane connections from Branch 2 to Branch 3.

Example 7-21 *All VPNs Are Directly Connected Between Branch 2 and Branch 3*

```
!
```

Technet24

```
! BR2-Edge1 is one hop away from Branch 3 in VR
!
BR2-Edge1# traceroute vrf 10 10.10.103.1
<<<Output omitted for brevity>>>
  1 BIZ-INET-BR3-Edge1 (209.165.201.8) 2 msec 1
  2 BR3-Edge1-Core (172.16.103.2) 1 msec 2 msec
BR2-Edge1#
!
! BR2-Edge1 is one hop away from Branch 3 in VR
!
BR2-Edge1# traceroute vrf 20 10.2.103.1
<<<Output omitted for brevity>>>
  1 BIZ-INET-BR3-Edge1 (209.165.201.8) 2 msec 1
  2 BR3-Edge1-Core (172.20.103.2) 2 msec 2 msec
BR2-Edge1#
```

Example 7-22 highlights the necessary changes to the centralized control policy. In order to enact this policy, a new criterion was added to sequence 41 so that it only applies to routes from the corporate VPN. Now sequence 41 will match all routes from European branches in VPN 10 and accept them without any modifications. In addition, sequence 51 is added to match the routes from European branches in the PCI VPN. These routes have their TLOCs (next-hop attributes) altered so that the prefixes are advertised as being reachable from the European data centers.

Example 7-22 *Use Case 8: Multi-Topology Policy*

```
!
! In order to create different logical topologi
! routes need to be manipulated on a per-VPN ba
! done in sequence 41, which matches and accept
! VPN, and in sequence 51, which matches routes
! TLOC list with the TLOCs of DC2.
!
policy
 control-policy Euro_Reg_Mesh_with_FW_MultiTopo
    sequence 1
     match tloc
      site-list DCs
     !
     action accept
     !
    !
    sequence 11
     match tloc
      site-list Europe_Branches
     !
     action accept
     !
```

Technet24

```
    !
    sequence 21
     match route
      prefix-list Default_Route
      site-list Europe_DC
     !
     action accept
      set
       preference 100
      !
     !
    !
    sequence 31
     match route
      site-list DCs
      prefix-list _AnyIpv4PrefixList
     !
     action accept
     !
    !
    sequence 41
     match route
      site-list Europe_Branches
     vpn-list CorporateVPN
      prefix-list _AnyIpv4PrefixList
```

```
 !
 action accept
 !
!
sequence 51
 match route
  site-list Europe_Branches
 vpn-list PCI_VPN
  prefix-list _AnyIpv4PrefixList
 !
 action accept
  set
  tloc-list Europe_DC_TLOCs
  !
 !
!
sequence 61
 match route
  site-list North_America_Branches
  prefix-list _AnyIpv4PrefixList
 !
 action accept
  set
   service  FW
  !
```

Technet24

```
      !
     !
  default-action reject
 !
 control-policy North_America_Reg_Mesh_with_FW
  ! <<<No changes made to this policy from Exam
brevity>>>
 !
 vpn-membership vpnMembership_-950781881
  ! <<<No changes made to this policy from Exam
brevity>>>
 !
 control-policy Set_DC_TLOC_Preference
  ! <<<No changes made to this policy from Exam
brevity>>>
 !
 lists
 ! <<<Some lists without changes from Example 7-
  !
  ! A new TLOC List is created for the TLOCs in
  !
  tloc-list Europe_DC_TLOCs
   tloc 10.0.20.1 color mpls encap ipsec
   tloc 10.0.20.1 color biz-internet encap ipse
   tloc 10.0.20.2 color mpls encap ipsec
```

```
    tloc 10.0.20.2 color biz-internet encap ipse
   !
  !
 !
 apply-policy
  site-list Europe_Branches
   control-policy Euro_Reg_Mesh_with_FW_MultiTop
  !
  site-list DCs
   control-policy Set_DC_TLOC_Preference in
  !
  site-list North_America_Branches
   control-policy North_America_Reg_Mesh_with_FW
  !
  site-list BranchOffices
   vpn-membership vpnMembership_-950781881
  !
 !
```

As the multi-topology policy in Example 7-22 shows, creating different logical topologies on a per-VPN basis involves manipulating the routes on a per-VPN basis. The centralized policy in this use case creates a new TLOC list with the TLOCs from the London data center and applies the TLOC list as the next hop for the routes in VPN 20, the PCI segment. The effect of this

Technet24

policy is that, while the connectivity in VPN 10 will continue to be direct, the communication in VPN 20 will be proxied through the data centers. You can see the effects of this policy in Example 7-23.

Example 7-23 *VPN 10 Has a Direct Data Plane; VPN 20 Must Transit the Data Center When Passing Traffic from Branch 2 to Branch 3*

```
!
! BR2-Edge1 is one hop away from Branch 3 in VR
BR2-Edge1# traceroute vrf 10 10.10.103.1
<<<Output omitted for brevity>>>
  1 BIZ-INET-BR3-Edge1 (209.165.201.8) 2 msec 1
  2 BR3-Edge1-Core (172.16.103.2) 1 msec 2 msec

BR2-Edge1#
!
! BR2-Edge1 is additional hops away from Branch
!
BR2-Edge1# traceroute vrf 20 10.20.103.1
<<<Output omitted for brevity>>>
  1 BIZ-INET-DC2-Edge1 (209.165.201.3) 2 msec 1
  2 BIZ-INET-BR3-Edge1 (209.165.201.8) 2 msec 2
  3 BR3-Edge1-Core (172.20.103.2 2) msec 2 msec
BR2-Edge1#
```

Use Case 8 Review

This use case shows how to create a multi-topology policy by combining the elements of prior use cases. This use case demonstrates how network administrators can create different data plane topologies on a per-segment basis to meet their business needs by using the VPN criterion in addition to other criteria to select, filter, and manipulate the control plane updates.

Use Case 9: Creating Extranets and Access to Shared Services

In the final centralized control policy use case, we will explore building an extranet. Enterprises commonly use extranets to provide multiple business partners with access to a shared resource, such as an ERP solution, while at the same time ensuring that the partners cannot directly access each other. In this use case, we are going to use VPNs 101 and 102, at Sites 101 and 102, respectively, to represent the business partners. The shared resource that these partners need to access will be in VPN 100 at DC1. Figure 7-77 shows the extranet that will be built on top of the existing policies from the previous examples.

Technet24

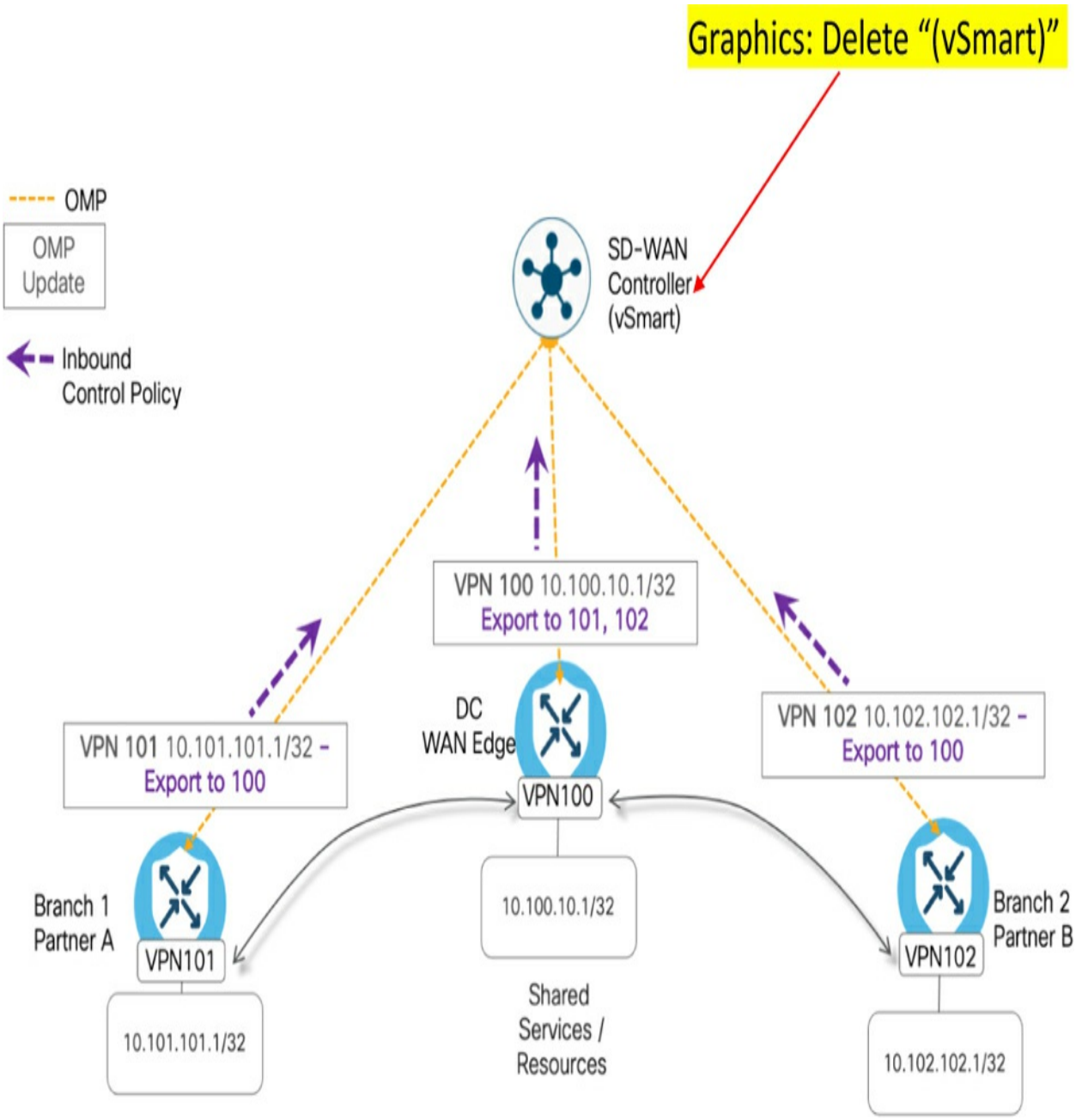


Figure 7-77 *Building an Extranet to Enable Partner Connectivity*

In order to build the necessary connectivity to accomplish this design, there are several key parts of the centralized policy that need to be either adjusted or created from scratch. The VPN membership policy created in Use Case 7 needs to be adjusted to account for the new VPNs being used for the extranet.

In addition, the inbound route policies need to be either edited (in the case of the data centers) or created (in the case of the branches) so that the route leaking via the **export-to** command can take effect. The Set_DC_TLOC_Preference policy has to be rewritten to include the statements for route leaking. Because there can be only a single control policy for a specific site ID applied in a single direction, the functions of setting the TLOC preferences and leaking the routes from VPN 100 to VPN 101 and 102 need to be combined into a single policy. We will rename this new policy DC_Inbound_Control_Policy to reflect the fact that its purpose is broader than just manipulating the data center TLOCs.

The policy in Example 7-24 implements the necessary changes to completely deploy this extranet. This example presents the policy in its entirety so that you can use it as a reference.

Example 7-24 *Use Case 9: Extranet*

```
policy
!
! The VPN membership policy is extended to acc
! that need to be advertised to form the extra
! these are grouped into the CLIENT_VPNS (VPNs
! SERVICE_VPN (VPN 100).
!
```

Technet24

```
vpn-membership vpnMembership_-1376283532
    sequence 10
     match
      vpn-list CLIENT_VPNS
     !
     action accept
     !
    !
    sequence 20
     match
      vpn-list CorporateVPN
     !
     action accept
     !
    !
    sequence 30
     match
      vpn-list SERVICE_VPN
     !
     action accept
     !
    !
    sequence 40
     match
      vpn-list PCI_VPN
```

```
      !
      action accept
      !
     !
  default-action reject
 !
!
! The former "Set_DC_TLOC_Preference" policy ha
! "DC_Inbound_Control_Policy" and has had seque
perform
! the route leaking from VPN 100 to VPN 101 and
!
 control-policy DC_Inbound_Control_Policy
    sequence 1
     match tloc
      originator 10.0.10.1
     !
     action accept
      set
       preference 500
      !
     !
    !
    sequence 11
     match tloc
```

Technet24

```
        originator 10.0.10.2
       !
       action accept
        set
         preference 400
        !
       !
      !
      sequence 21
       match tloc
        originator 10.0.20.1
       !
       action accept
        set
         preference 500
        !
       !
      !
      sequence 31
       match tloc
        originator 10.0.20.2
       !
       action accept
        set
         preference 400
```

```
      !
     !
    !
    sequence 41
    match route
     vpn-list SERVICE_VPN
      prefix-list _AnyIpv4PrefixList
     !
    action accept
     export-to vpn-list CLIENT_VPNS
      set
      !
      ! An OMP TAG is similar to a route tag th
      ! routing protocols.  While it is not str
TAG
      ! during redistribution, it may become us
with
      ! tracking how routes are propagating as
      ! criteria to filter on.
      !
       omp-tag 100
      !
     !
    !
  default-action accept
```

Technet24

```
 !
!
! The "North_America_Reg_Mesh_with_FW" policy m
! versions.
!
 control-policy North_America_Reg_Mesh_with_FW
    sequence 1
     match tloc
      site-list DCs
     !
     action accept
     !
    !
    sequence 11
     match tloc
      site-list North_America_Branches
     !
     action accept
     !
    !
    sequence 21
     match route
      prefix-list Default_Route
      site-list North_America_DC
     !
```

```
      action accept
       set
        preference 100
       !
      !
    !
    sequence 31
     match route
      site-list DCs
      prefix-list _AnyIpv4PrefixList
     !
     action accept
     !
    !
    sequence 41
     match route
      site-list North_America_Branches
      prefix-list _AnyIpv4PrefixList
     !
     action accept
     !
    !
    sequence 51
     match route
      site-list Europe_Branches
```

Technet24

```
       prefix-list _AnyIpv4PrefixList
      !
      action accept
       set
        service  FW
       !
      !
     !
   default-action reject
  !
  !
  ! The "Euro_Reg_Mesh_with_FW_MultiTopo" policy
previous
  ! versions.
  !
 control-policy Euro_Reg_Mesh_with_FW_MultiTopo
     sequence 1
      match tloc
       site-list DCs
      !
      action accept
      !
     !
     sequence 11
      match tloc
```

```
        site-list Europe_Branches
       !
       action accept
       !
      !
      sequence 21
       match route
        prefix-list Default_Route
        site-list Europe_DC
       !
       action accept
        set
         preference 100
        !
       !
      !
      sequence 31
       match route
        site-list DCs
        prefix-list _AnyIpv4PrefixList
       !
       action accept
       !
      !
      sequence 41
```

Technet24

```
      match route
       site-list Europe_Branches
       vpn-list CorporateVPN
       prefix-list _AnyIpv4PrefixList
      !
      action accept
      !
     !
     sequence 51
      match route
       site-list Europe_Branches
       vpn-list PCI_VPN
       prefix-list _AnyIpv4PrefixList
      !
      action accept
       set
        tloc-list Europe_DC_TLOCs
       !
      !
     !
     sequence 61
      match route
       site-list North_America_Branches
       prefix-list _AnyIpv4PrefixList
      !
```

```
     action accept
      set
       service  FW
      !
     !
    !
 default-action reject
!
!
! A new control policy is created in order to
! branch sites and export the routes from VPN
! An OMP tag is again added during the route i
! required, it is highly recommended to do so.
!
control-policy Branch_Extranet_Route_Leaking
   sequence 1
    match route
     vpn 101
     prefix-list _AnyIpv4PrefixList
    !
    action accept
       export-to vpn-list SERVICE_VPN
       set
        omp-tag 101
     !
```

Technet24

```
     !
    !
    sequence 11
       match route
       vpn 102
      prefix-list _AnyIpv4PrefixList
     !
     action accept
        export-to vpn-list SERVICE_VPN
        set
          omp-tag 102
        !
     !
    !
 default-action accept
!
lists
 prefix-list Default_Route
  ip-prefix 0.0.0.0/0
 !
 site-list BranchOffices
  site-id 100-199
 !
 site-list DCs
  site-id 10-50
```

```
    !
    site-list Europe_Branches
     site-id 102-103
    !
    site-list Europe_DC
     site-id 20
    !
    site-list North_America_Branches
     site-id 101
    !
    site-list North_America_DC
     site-id 10
  !
    tloc-list Europe_DC_TLOCs
     tloc 10.0.20.1 color mpls encap ipsec
     tloc 10.0.20.1 color biz-internet encap ipse
     tloc 10.0.20.2 color mpls encap ipsec
     tloc 10.0.20.2 color biz-internet encap ipse
    !
    vpn-list CLIENT_VPNS
     vpn 101
     vpn 102
    !
    vpn-list CorporateVPN
     vpn 10
```

Technet24

```
   !
   vpn-list PCI_VPN
    vpn 20
   !
   vpn-list GUEST_VPN
    vpn 50
   !
   vpn-list SERVICE_VPN
    vpn 100
   !
   prefix-list _AnyIpv4PrefixList
    ip-prefix 0.0.0.0/0 le 32
   !
  !
 !
 !
  ! Lastly, all of the policies are applied.  No
  ! route leaking must be applied inbound.
  !
apply-policy
 site-list Europe_Branches
  control-policy Euro_Reg_Mesh_with_FW_MultiTop
 !
 site-list DCs
  control-policy DC_Inbound_Control_Policy in
```

```
  !
  site-list BranchOffices
   control-policy Branch_Extranet_Route_Leaking
   vpn-membership vpnMembership_-1376283532
  !
  site-list North_America_Branches
   control-policy North_America_Reg_Mesh_with_FW
  !
 !
```

In Example 7-24, route leaking is configured in both the Branch_Extranet_Route_Leaking and the DC_Inbound_Control_Policy policies with the **export-to** action. The **export-to** action allows the matching routes to be copied from the existing VPN and injected into a different VPN or VPN list, using the syntax **export-to** {**vpn** *vpn-id* | **vpn-list** *vpn-list*}.



Route leaking is a powerful and flexible tool that network administrators have at their disposal. However, there are some considerations with how the **export-to** command must be used. First, the **export-to** action must be applied on an inbound control policy. If the **export-to** action is applied in the outbound direction, the action will have no effect. Second, the **export-to** command only works to leak routes from a service-side VPN into a different service-side VPN. The **export-to** command cannot be used to leak routes into

Technet24

or out of VPN 0 or VPN 512. The isolation of these VPNs is intentional and, specifically with VPN 0, has significant security implications and needs to be executed carefully. There is some discussion in Chapter 5, "Cisco Catalyst SD-WAN Design and Migration," about different design concepts that aim to solve these challenges.

Note

The method described here is effective for service VPN-to-service VPN route leaking. You can achieve bidirectional route leaking between a service VPN and a global VPN by using the **route-replicate** and **redistribute** commands in conjunction; together, these commands allow you to share common services that multiple data segments need to access. For more comprehensive information, consult the Cisco Catalyst SD-WAN Routing Configuration Guide.

When the policy in Example 7-24 is applied, the results of the route leaking can be seen in the routing tables on BR1-Edge1 and BR2-Edge1, as shown in Figure 7-78 and Figure 7-79, respectively. The BR1-Edge1 and BR2-Edge1 routing tables have installed the two routes for the services VPN. These routes are also displayed with the tag 100 as a reminder that they originated in VPN 100.

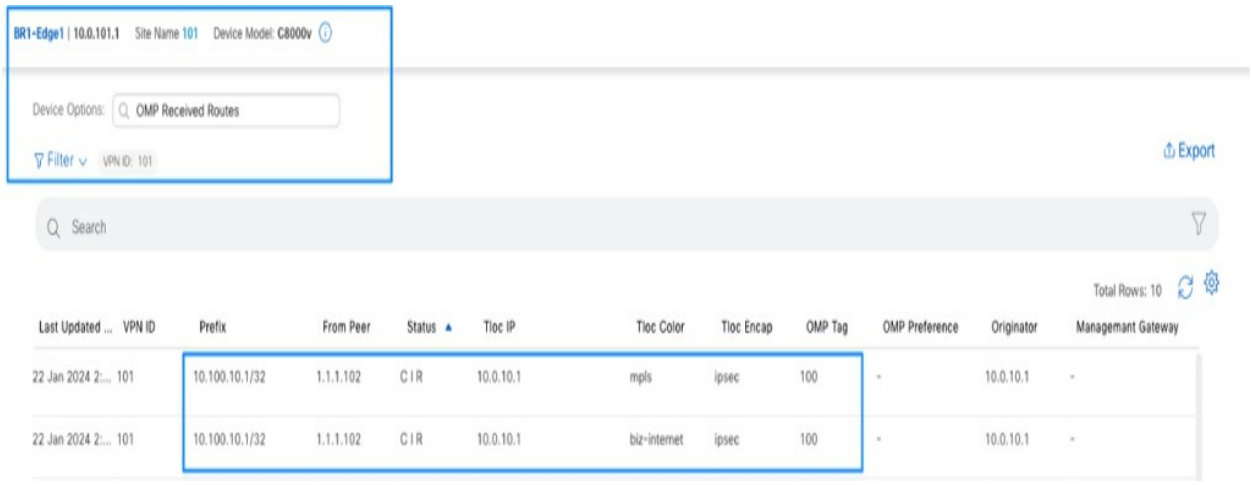


Figure 7-78 *Service Routes from VPN 100 Have Been Imported into VPN 101*

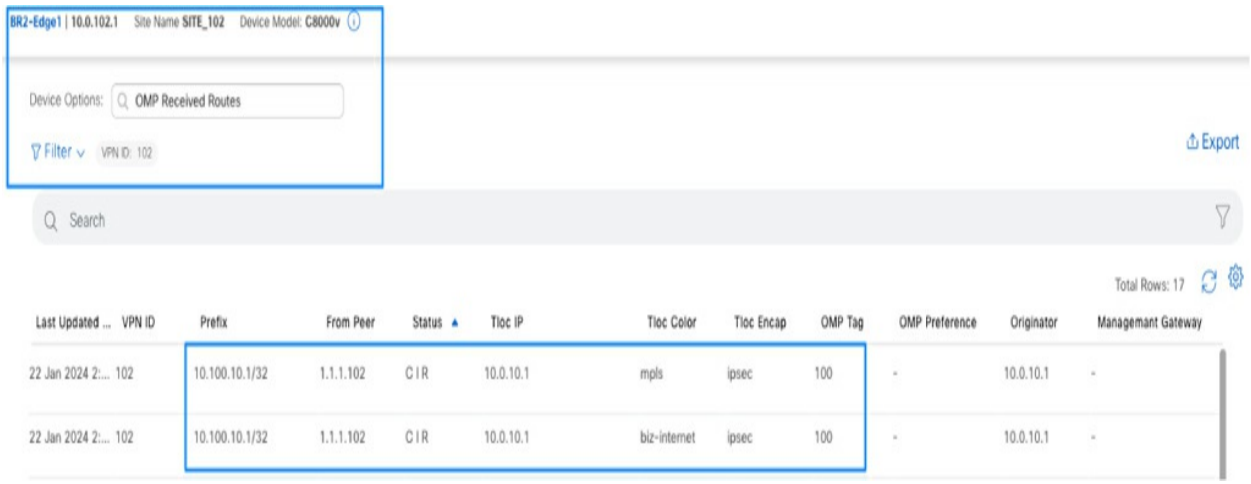


Figure 7-79 *Service Routes from VPN 100 Have Been Imported into VPN 102*

Finally, the data plane of the extranet can be validated by ensuring that VPN 101 and VPN 102 both have access to the shared services hosted in VPN 100 but that VPN 101 cannot communicate directly with VPN 102. Example 7-25 shows the results from the pings to test the data plane of the extranet.

Example 7-25 *Validating the Data Plane Connectivity of the Extranet*

Technet24

Connections

```
!
! BR2-Edge1 is able to reach the shared service
!
BR2-Edge1#
BR2-Edge1# ping vrf 102 10.100.10.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.100.10.1,
!!!!!
Success rate is 100 percent (5/5), round-trip r
!
! BR2-Edge1 is not able to directly reach hosts
!
BR2-Edge1# ping vrf 102 10.101.101.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.101.101.1,
.....
Success rate is 0 percent (0/5)
!
!
! BR1-Edge1 is able to reach the shared service
!
BR1-Edge1# ping vrf 101 10.100.10.1
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.100.10.1,
!!!!!
Success rate is 100 percent (5/5), round-trip r
!
! BR1-Edge1 is not able to directly reach hosts
!
BR1-Edge1# ping vrf 101 10.102.102.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.102.102.1,
.....
Success rate is 0 percent (0/5)
BR1-Edge1#
```

The output in Example 7-25 confirms that the extranet is working as expected. VPNs 101 and 102 have access to the shared resource that exists in VPN 100, but VPN 101 and 102 are not able to communicate between themselves.

Use Case 9 Review

This use case shows how to create an extranet to allow business partners to communicate while still maintaining security and isolation across the SD-WAN fabric. This extranet is configured by establishing a unique VPN for each of the different entities and then leaking routes from the client VPNs

Technet24

(101 and 102) into the service VPN (100) and vice versa. Route leaking with centralized control policies is a technique that can be used for many different purposes. In addition to creating extranets for shared services access, as was done in this use case, another common use case that requires VPN leaking is using the SD-WAN security feature set, particularly the zone-based firewall. Cisco SD-WAN security is covered in Chapter 11, "Cisco Catalyst SD-WAN Security."

Summary

This chapter discusses one of the key types of SD-WAN policies: centralized control policies applied using a traditional approach as well as topology groups. It includes several use cases using centralized control policies and shows how policies can be used to manipulate the structure of the SD-WAN data plane fabric, turning an SD-WAN fabric from a full-mesh deployment to a strict hub-and-spoke deployment or a regional mesh. In addition, this chapter provides several traffic engineering use cases that show how to use TLOC lists to manipulate the next-hop information of an OMP route, as well as network service insertion. Finally, this chapter shows how to use centralized control policies to harden the security posture of the network, including using VPN membership policies and building extranets with route leaking.

This chapter shows how various kinds of policies can be combined and used together to create the architecture necessary to meet any business objective in

an easy and intuitive but powerful way. Administrators are advised to select either the new topology approach or use centralized control policies to avoid unnecessary complexity. Using the topology approach can enhance configuration reusability and provide a more structured approach, potentially simplifying network operations.

Review All Key Topics

Review the most important topics in the chapter, noted with the Key Topic icon in the outer margin of the page. Table 7-2 lists these key topics and the page number on which each is found.



Table 7-2 *Key Topics*

Technet24

Key Topic Element	Description	Page
Paragraph	Data plane tunnels	
Paragraph	BFD session status and data plane tunnel status	
Paragraph	TLOC attribute in OMP routes.	
Paragraph, Example 7-8	OMP route status codes	
Table 7-1	OMP status codes	
Paragraph, Figure 7-65	TLOC Preference value	
Paragraph	Weight attribute for configuring unequal-cost load-sharing	
Figure 7-71, Paragraph	Routes with greatest OMP Preference values	
Paragraph	VPN membership policies	
Paragraph	Route leaking	

Define Key Terms

Define the following key terms from this chapter and check your answers in the glossary:

extranet

inbound control policy

multi-topology

originator

outbound control policy

service insertion

TLOC list

topology group

Chapter Review Questions

1. What is the default setting of the default action in a centralized control policy?

a. Accept

Technet24

b. Permit

c. Reject

d. Deny

e. There is no default action; an action must be configured manually.

2. Which of the following are configuration options for a TLOC list? Choose 4.

a. Site ID

b. System IP address

c. Color

d. Encapsulation

e. Preference

f. Weight

g. VPN

h. Prefix

3. The TLOC attribute Weight is used _____.

a. as the first and most important criterion in the OMP best-path selection process.

b. as the final tie-breaker in the OMP best-path selection process.

c. for determining the ratio of flows for load-sharing on the TLOCs that have been selected as the best paths.

d. for turning off the anti-gravity machine.

4. True or false: OMP Route Preference values can be configured via feature templates and device templates.

a. True

b. False

5. What is the status of an OMP process that has been inserted into the IP routing table?

a. C I R

b. C R

c. R

d. Rej, R, Inv

Technet24

6. Preference is an OMP attribute associated with which of the following?

a. TLOCs

b. OMP routes

c. Both TLOCs and OMP routes

d. Neither TLOCs nor OMP routes

7. What are VPN membership policies used to do?

a. Determine which users belong to which VPNs.

b. Determine which routes belong to which VPNs.

c. Determine which WAN Edge routers belong to which SD-WAN fabrics.

d. Determine which VPNs will be permitted to join the overlay fabric on a WAN Edge router.

e. Determine the ratio of flows for load-sharing on the TLOCs that have been selected as the best paths.

8. True or false: Centralized control policies that leak routes must always be applied in the outbound direction.

a. True

b. False

9. True or false: A centralized control policy can be used to leak routes between service-side VPNs and VPN 0.

a. True

b. False

10. Which type of policy is used to export OMP routes from one VPN to another VPN?

a. Route import/export policies

b. VPN membership policies

c. Centralized control policies

d. Localized control policies

e. No policy; it is not possible to have the same route in more than one VPN.

11. True or false: Topology groups help to achieve the same behavior as centralized control policies.

a. True

b. False

Technet24

References

“Cisco Catalyst SD-WAN Command Reference,” https://www.cisco.com/c/en/us/td/docs/routers/sdwan/command/sdwan-cr-book.html.

“Policy Groups Configuration Guide, Cisco IOS XE Catalyst SD-WAN Release 17.x,” https://www.cisco.com/c/en/us/td/docs/routers/sdwan/configuration/Policy-Groups/policy-groups/m-topology.html.

“Hub-and-Spoke Configuration, Cisco Catalyst SD-WAN Routing Configuration Guide,”

https://www.cisco.com/c/en/us/td/docs/routers/sdwan/configuration/routing/ios xe-17/routing-book-xe/hub-and-spoke.html

“Network Hierarchy and Resource Management, Systems and Interfaces Configuration Guide,” https://www.cisco.com/c/en/us/td/docs/routers/sdwan/configuration/system-interface/ios-xe-17/systems-interfaces-book-xe-sdwan/network-hierarchy-and-resource-management.html

“Route Leaking between VPNS, Cisco Catalyst SD-WAN Routing Configuration Guide,” https://www.cisco.com/c/en/us/td/docs/routers/sdwan/configuration/routing/io

xe-17/routing-book-xe/m-routing-leaking-for-service-sharing.html

Technet24

Chapter 8. Centralized Data Policies

This chapter covers the following topics:

• **Centralized Data Policy Overview:** This section reviews the basics of centralized data policies and the directionality of policies when applied to a WAN Edge router.

• **Use Case 10: Direct Internet Access for Guest Users:** This section covers a simple data policy to ensure that only publicly routable packets are forwarded to the Internet. This section also discusses how to build a centralized data policy with the SD-WAN Manager GUI.

• **Use Case 11: Direct Cloud Access for Trusted Applications:** This section covers the use of a centralized data policy to change the forwarding path for a specific application and provide direct Internet egress.

• **Use Case 12: Application-Based Traffic Engineering:** This section reviews different methods for performing traffic engineering of flows that traverse the SD-WAN fabric.

• **Use Case 13: Protecting Corporate Users with a Secure Internet Gateway:** This section reviews how to redirect traffic to Cisco Umbrella Secure Internet Gateway.

• **Use Case 14: Protecting Applications from Packet Loss:** This section

covers two different methodologies for reconstructing data flows that have been affected by lossy transport networks.

Building on the examples in Chapter 7, "Centralized Control Policies," this chapter focuses on centralized data policies. Unlike centralized control policies, which manipulate the routing information in the data plane, centralized data policies are used to override the forwarding decisions that would normally be made by a router and instead tell the router to follow a specific set of forwarding instructions. Such a policy can be implemented for a specific flow or application or for all of the traffic at a site. Because of the incredible control they provide to network administrators, centralized data policies are often referred to as "policy-based routing on steroids." However, centralized data policies are much easier to administer than traditional policy-based routing because, as their name indicates, they are all configured and applied centrally.

Centralized Data Policy Overview

A centralized data policy is a powerful tool that allows administrators to override the normal forwarding actions that would occur in the data plane and specify a different set of actions that should be taken instead. As the use cases in this chapter show, those new actions could be as simple as dropping a packet, redirecting a flow down a specific path or to a specific service, providing additional data plane services to accelerate data transfers and protect against packet loss, or any combination of these actions.

Technet24



As discussed in Chapter 6, "Introduction to Cisco Catalyst SD-WAN Policies," each type of policy has a specific directionality. Centralized data policies can either be applied to traffic that is originating "from tunnel," traffic that is originating "from service," or traffic that is traversing a WAN Edge router in both directions. Figure 8-1 illustrates these directions.

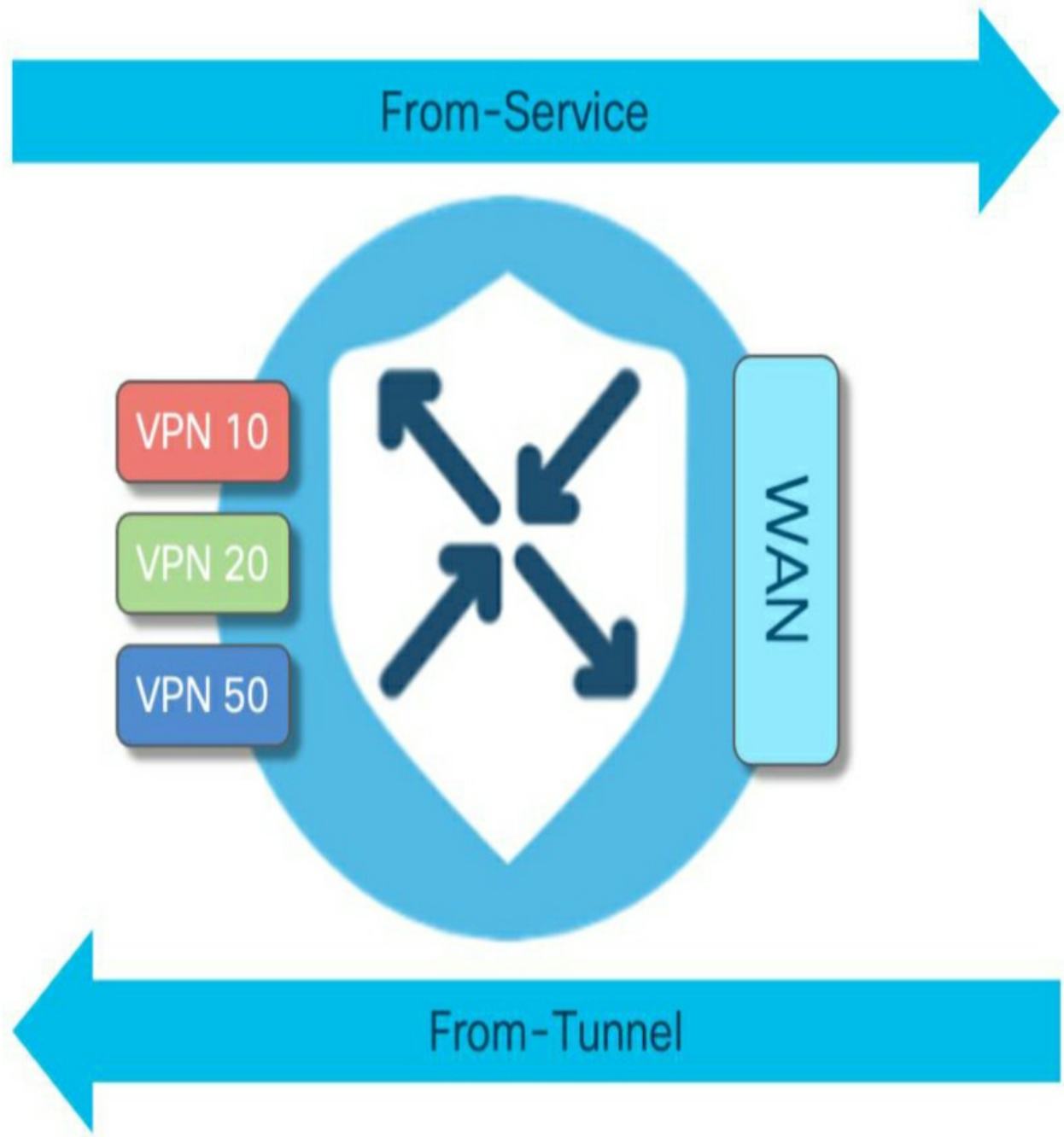


Figure 8-1 *Data Policy Directionality*

The following sections examine several different sets of common business requirements and explore how network administrators can use centralized data policies to handle them. These use cases highlight common applications

Technet24

of centralized data policies and identify many of the building blocks of centralized data policies from which network administrators can build their own policies to accomplish business objectives. The examples in this chapter build on the network and the examples provided in Chapter 7. Figure 8-2 shows the topology used with all the use cases throughout this chapter, and Example 7-24 at the end of Chapter 7 provides the centralized policy that is used as a starting point.

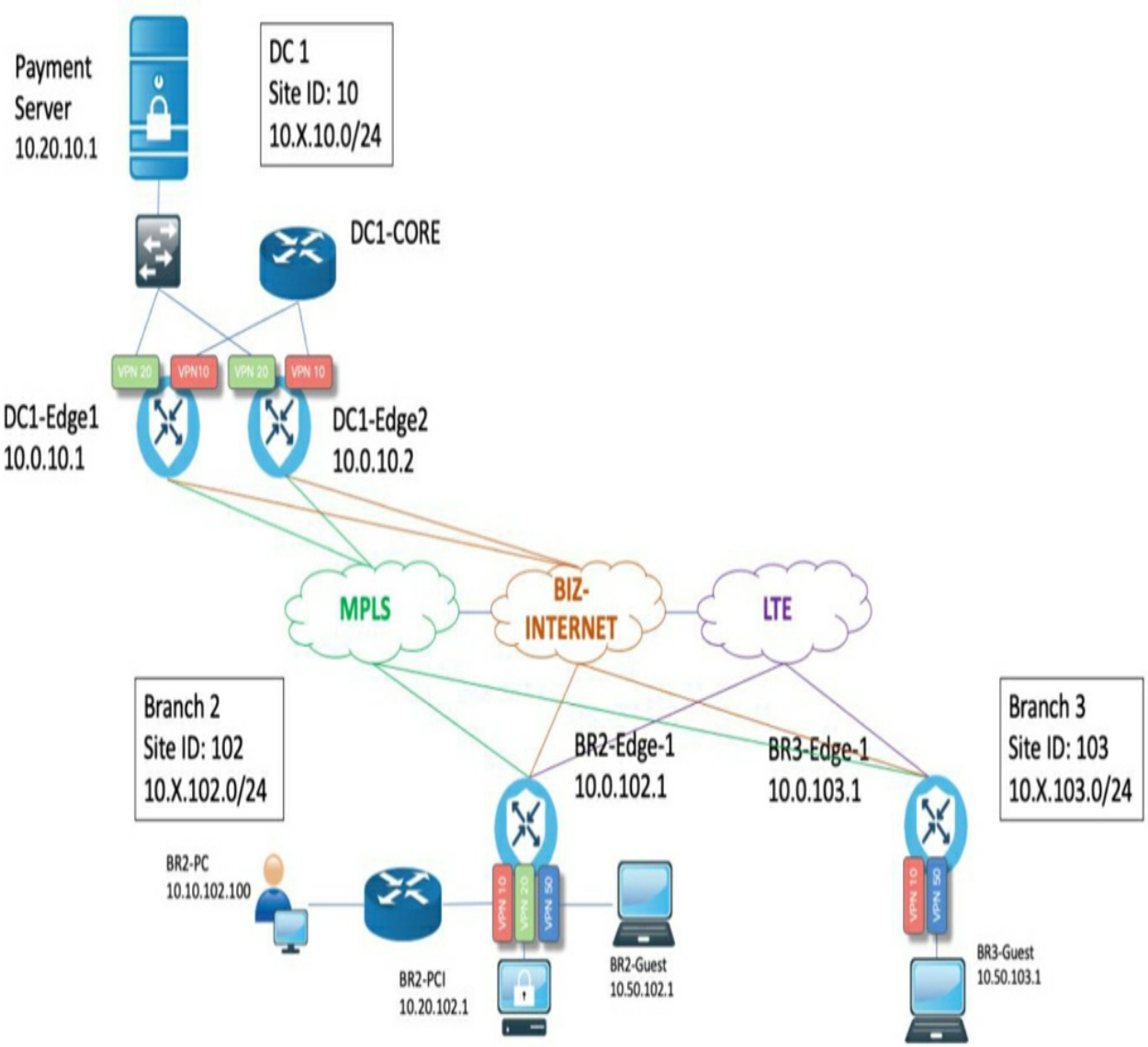


Figure 8-2 *Network Topology for Chapter 8*

Although the network featured in Figure 8-2 is identical to the network used in Chapter 7, there are several changes in this diagram that accompany some elements that are used throughout this chapter. Specifically, Branch 2 is drawn in greater detail to illustrate the end hosts that reside in the Corporate, PCI, and Guest data segments (that is, the service VPNs). Also note that the payment processing server is located in Data Center 1, in the PCI VPN. As noted in Chapter 7, the service-side addressing in this network follows the 10.X.Y.0/24 structure, where X signifies the service-side VPN, and Y signifies the site ID. The first few use cases in this chapter focus on VPN 10 and VPN 50, and service-side addressing is in the 10.10.Y.0/24 and 10.50.Y.0/24 address blocks. This topology will allow us to explore different types of policies, how they interact with the SD-WAN fabric, and how network administrators can apply these policies to solve business problems.

Use Case 10: Direct Internet Access for Guest Users

This use case builds on the VPN membership policy we explored in Chapter 7 to provide Internet access to guest users. In Chapter 7, Example 7-20 shows that users in the Guest VPN on BR2-Edge1 are able to reach the Internet but are not able to reach the other branches. But this is not the complete story, as all the traffic in the VPN is following the default route. Even traffic that is destined to addresses that are not reachable across the public Internet, such as traffic using private (RFC 1918) addresses, is being forwarded out onto the

Technet24

public Internet, where it is eventually dropped, as shown in Example 8-1.

Example 8-1 *Connectivity from the Guest VPN on BR2-Edge1*

```
! BR2-Edge1 is able to reach the public interne

BR2-Edge1# ping vrf 50 208.67.222.222
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 208.67.222.2
!!!!!
Success rate is 100 percent (5/5), round-trip r
```

```
! BR2-Edge1 is unable to reach BR3-Edge1 in the

BR2-Edge1# ping vrf 50 10.50.103.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.50.103.1,
.....
Success rate is 0 percent (0/5)
!
! Looking at the traceroute to 10.50.103.1 and
! that BR2-Edge1 is using the default route to
```

```
! packet, rather than dropping the traffic.
!
BR2-Edge1# traceroute vrf 50 10.50.103.1
Type escape sequence to abort.
Tracing the route to 10.50.103.1
VRF info: (vrf in name/id, vrf out name/id)
  1 209.165.201.30 1 msec 1 msec 0 msec
  2 198.18.128.1 2 msec 2 msec 1 msec
  3 10.255.0.3 1 msec 1 msec 1 msec
  4 10.1.27.9 1 msec 2 msec 1 msec
  5 64.100.12.36 2 msec 2 msec 2 msec
  6 64.102.244.193 3 msec 3 msec 3 msec
  7 64.102.244.177 2 msec 2 msec 2 msec
  8 64.102.254.210 1 msec 2 msec 2 msec
  9  *  *  *
 10  *  *  *
! Output omitted for brevity
29 * * *
30 * * *
!
! The output indicates that this traffic is be
! to the transit interfaces in VPN 0 straight t
!
BR2-Edge1# show sdwan policy service-path vpn 5
10.50.102.1 dest-ip 10.50.103.1  protocol 1 al
```

Technet24

```
  Next Hop: IPsec
    Source: 192.168.20.6 12346 Destination: 192.1
  Remote Color: mpls Remote System IP: 10.0.10.1
  Next Hop: IPsec
    Source: 209.165.201.6 12346 Destination: 209.
  biz-internet Remote Color: biz-internet Remote
  Next Hop: IPsec
    Source: 209.165.202.130 12346 Destination: 20
  Remote Color: biz-internet Remote System IP: 10
```

Note

Public transport interfaces are commonly used not only to establish IPsec tunnels but also for providing Direct Internet Access breakout. If default routes from data centers are not present, other methods of providing Internet access, such as selective Direct Internet Access policy, are available. This approach is explained later in this chapter.

As Example 8-1 shows, bandwidth is being wasted because packets are being forwarded out the Internet interfaces that are destined to private, internal addresses and therefore cannot possibly reach their destinations across the Internet. Rather than uselessly forwarding these packets out to the Internet,

we could construct a centralized data policy to drop them.

The first step in the process of creating a centralized data policy is to create the data prefix lists that are necessary. The Custom Options menu for creating lists is in the upper-right corner of the Configuration > Policies screen (see Figure 8-3).

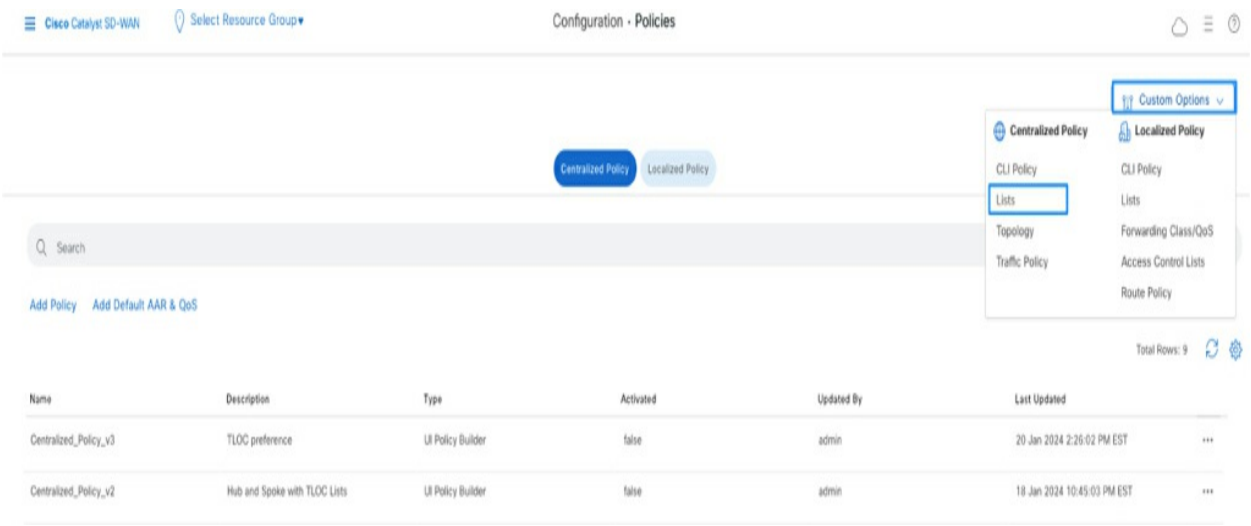


Figure 8-3 *Creating Required Lists in a Centralized Policy*

The first list that is needed for this centralized data policy is a data prefix list of bogon addresses. The following definition is from RFC 3871:

A "bogon" (plural: "bogons") is a packet with an IP source address in an address block not yet allocated by IANA or the Regional Internet Registries (ARIN, RIPE, APNIC[el]) as well as all addresses reserved for private or special use by RFCs. See [RFC3330] and [RFC1918].

Filtering out packets that are destined to networks that cannot exist on the

Technet24

public Internet will help save resources by avoiding forwarding unnecessary packets. You can create a data prefix list to accomplish this filtering by selecting the Data Prefix list type and clicking the blue +New Data Prefix List button. In the Data Prefix List configuration window that appears, you configure a list name and the necessary data prefixes, as shown in Figure 8-4.



Figure 8-4 *Configuring a Data Prefix Lists*

As Figure 8-4 indicates, this data prefix list includes RFC 1918 ranges, the 127.0.0.0/8 localhost range, and the 100.64.0.0/10 carrier-grade NAT range. None of these ranges are publicly routable, and therefore traffic that is destined to any of these destinations should not be forwarded to the Internet. To build a centralized policy to accomplish this filtering, the next step would be to build the data policy itself. Start by selecting Traffic Policy from the Custom Options drop-down menu (refer to Figure 8-3). In the Traffic Policy screen, you see three tabs across the top: Application Aware Routing, Traffic Data, and Cflowd. To create a new traffic data policy, select the Traffic Data

tab and then select the Create New option under the Add Policy menu (see Figure 8-5).



Figure 8-5 *Creating a New Centralized Data Policy*

Next, you need to add a sequence type to the new policy by clicking the +Sequence Type button. As shown in Figure 8-6, there are a number of different sequence types available in a centralized data policy, including Application Firewall, QoS, Service Chaining, and Traffic Engineering. These different sequence types expose a limited subset of the actions available to an administrator and highlight the different actions that may be used to address some of these use cases. The Custom sequence type provides access to all of the action options in the GUI, and many administrators prefer to use this sequence type for all use cases. For this example, select the Custom sequence type, as indicated in Figure 8-6.

Technet24

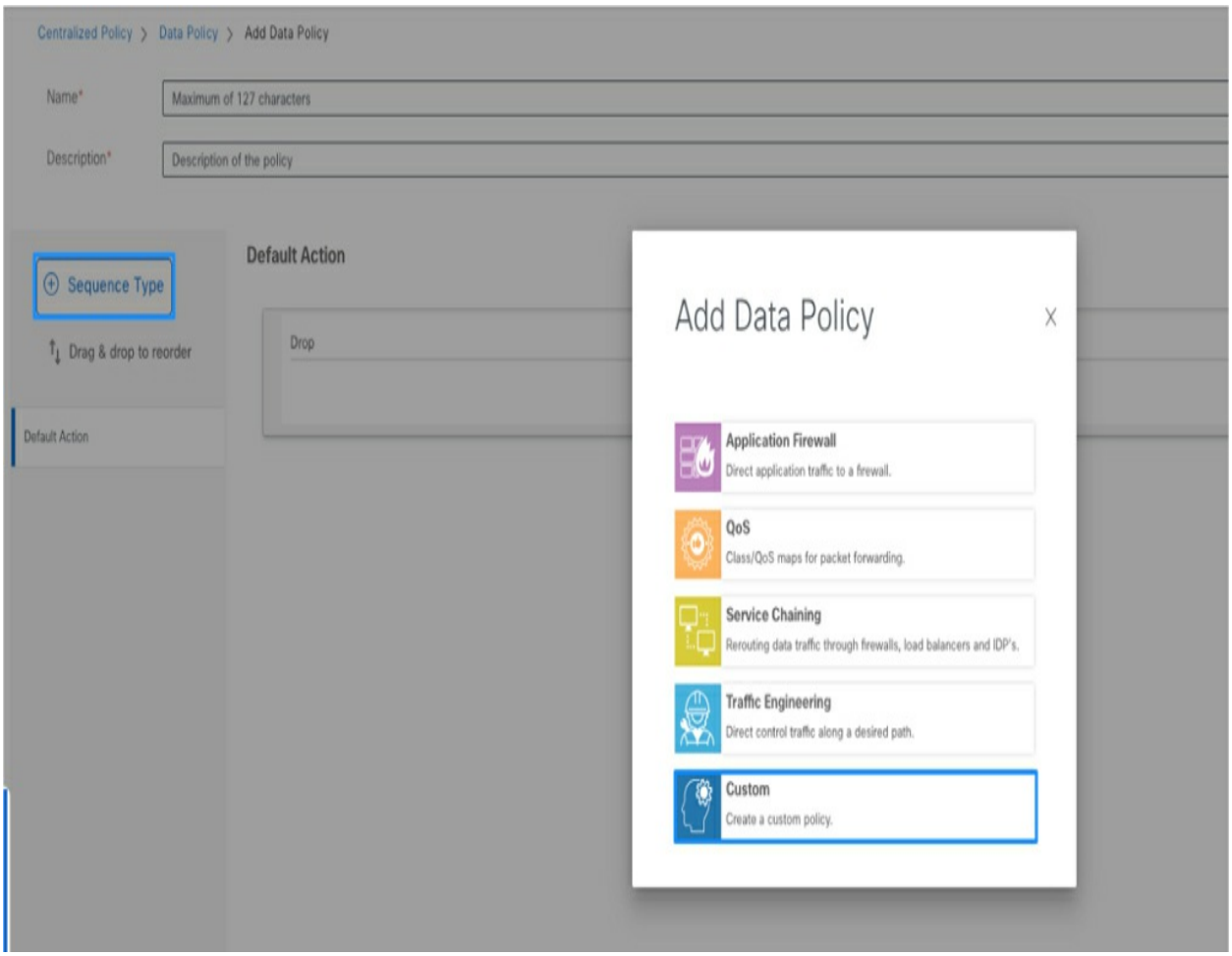


Figure 8-6 *Adding a New Sequence Type to the Data Policy*

This new policy needs to consist of two different sequence rules, as shown in Figure 8-7. The first sequence rule will match the packets destined to the addresses in the bogon data prefix list and drop them. There is an additional Counter action that is applied to this sequence rule as well. A counter named GUEST_DROPPED_PKTS is then specified to track the number of packets that match this sequence. While counters in and of themselves don't provide any impact to how traffic is forwarded, they do provide a useful tool when evaluating and troubleshooting policies.

The second sequence rule will send all other packets out the VPN 0 interfaces to the underlay. Matching all other packets is accomplished by not specifying any matching criteria at all in this sequence rule. There are three actions specified in the second sequence rule: NAT VPN, Local TLOC, and Counter. The NAT VPN action allows you to specify traffic that should be leaked to VPN 0 and forwarded out a NAT-enabled interface. The VPN argument for this action is always VPN 0. You cannot use the NAT VPN action to leak between service-side VPNs; the NAT VPN action can only be used to leak between a service VPN and VPN 0 for the purpose of allowing Direct Internet Access. The local TLOC action allow you to selectively route traffic to specific TLOCs, such as the NAT-enabled interfaces like the biz-internet TLOC in this case.

For ease of visibility, a counter called GUEST_DIA_PKTS is added to the **action** statements in this sequence that will count the number of packets forwarded by the second sequence.

Technet24

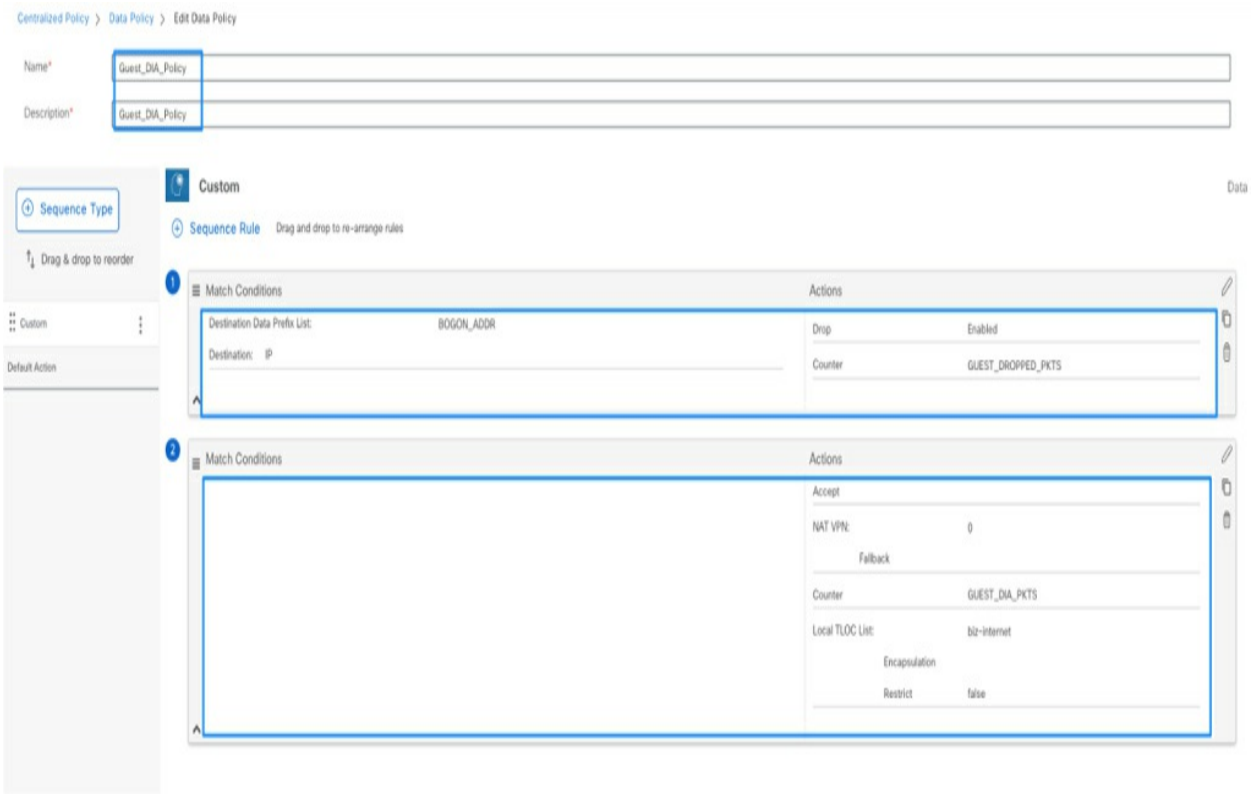


Figure 8-7 *Guest Internet Access Data Policy*

Once the necessary sequences have been added and a name and description have been added, the centralized data policy can be saved. The next step in the configuration process is to add this data policy to the existing centralized policy from Chapter 7. The easiest way to do so is to select the centralized policy and create a copy of it by selecting that option from the drop-down menu on the right side of the screen (see Figure 8-8).

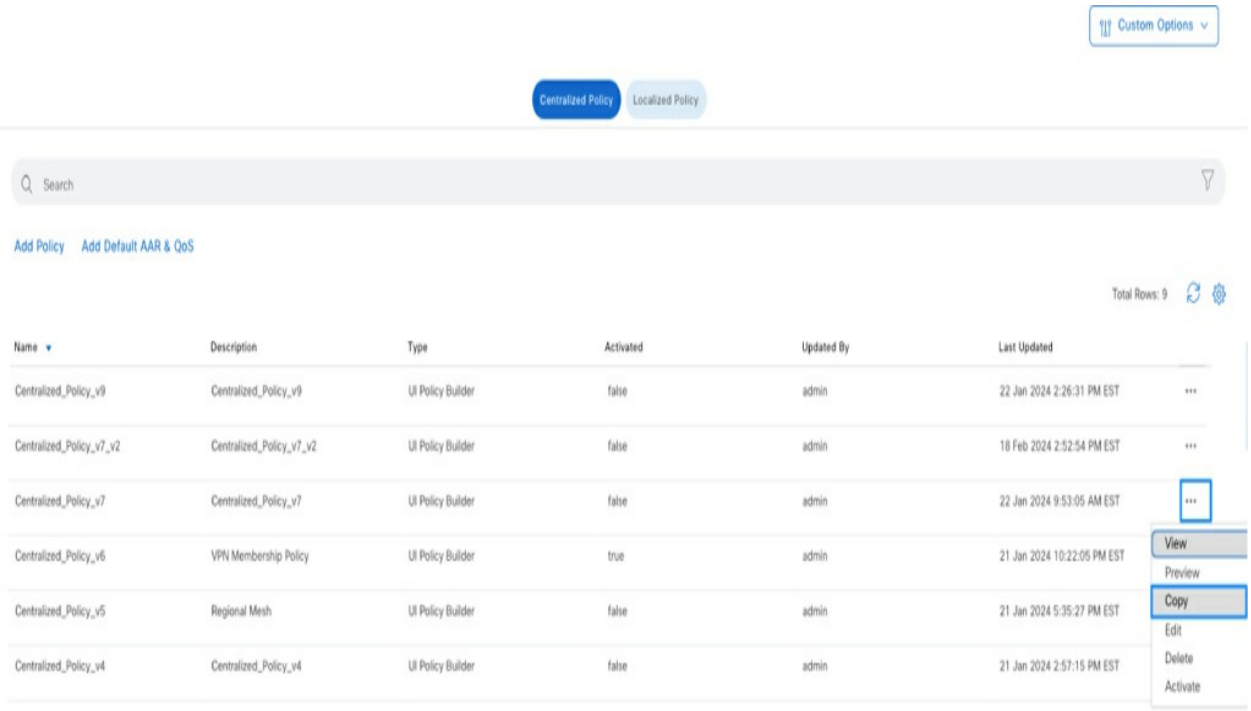


Figure 8-8 *Creating a Copy of a Centralized Policy*

Once the new policy is created, you can edit it by selecting the Edit option from the same drop-down menu on the newly copied policy. As shown in Figure 8-9, the Edit Policy screen has three main tabs at the very top: Policy Application, Topology, and Traffic Rules. You can select the Traffic Rules tab to reveal three sub-tabs: Application Aware Routing, Traffic Data, and Cflowd (the same tabs you saw when the data policy was initially created). After selecting the Traffic Data sub-tab, select the Add Policy menu in order to import the existing policy, as indicated in Figure 8-9.

Technet24

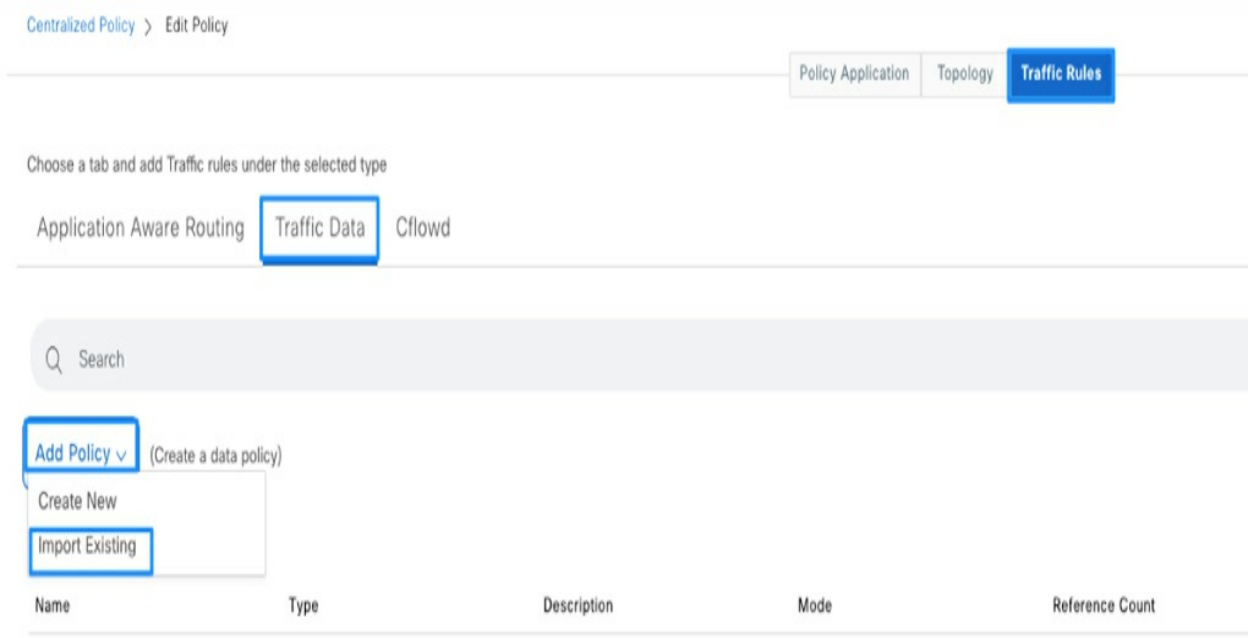


Figure 8-9 *The Edit Policy Screen*

When prompted, select the previously created Guest_DIA_Policy, as shown in Figure 8-10, and click the Import button.

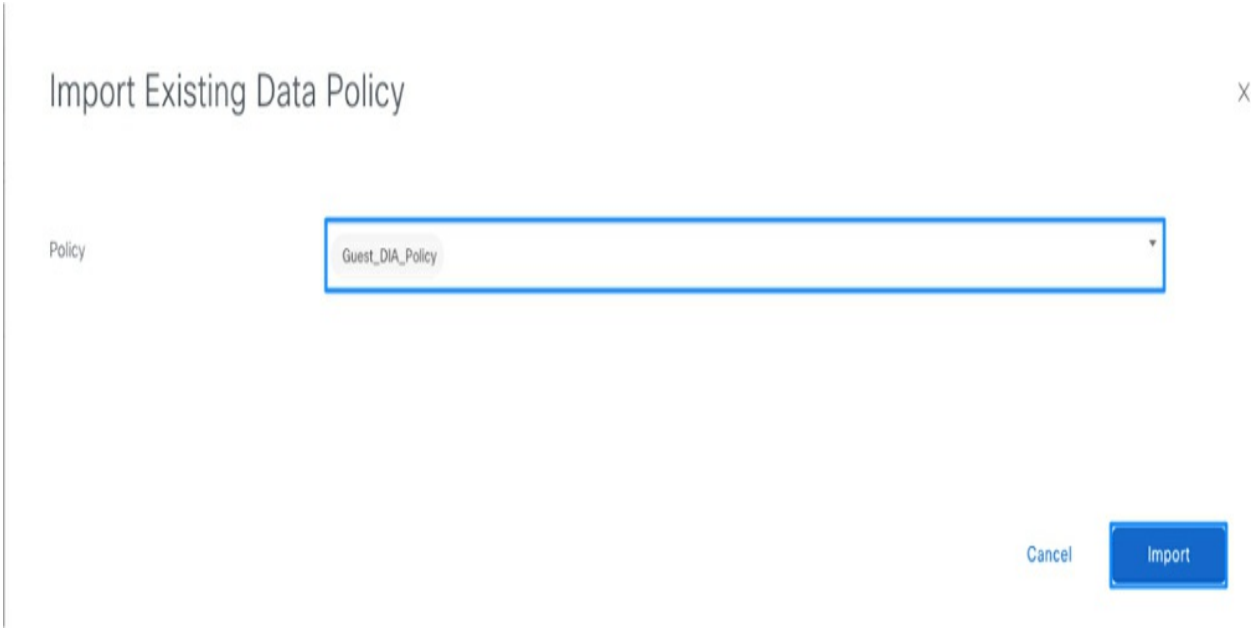


Figure 8-10 *Importing the Guest_DIA_Policy into the Centralized Policy*

Once the Guest DIA policy is referenced in the centralized policy, the last step in creating the centralized policy is to specify where the data policy should be applied by clicking the +New Site List and VPN List button. This policy is applied to the site ID list (BranchOffices), the VPN list (GUEST_ACCESS_VPN), and the direction (From Service), as shown in Figure 8-11.

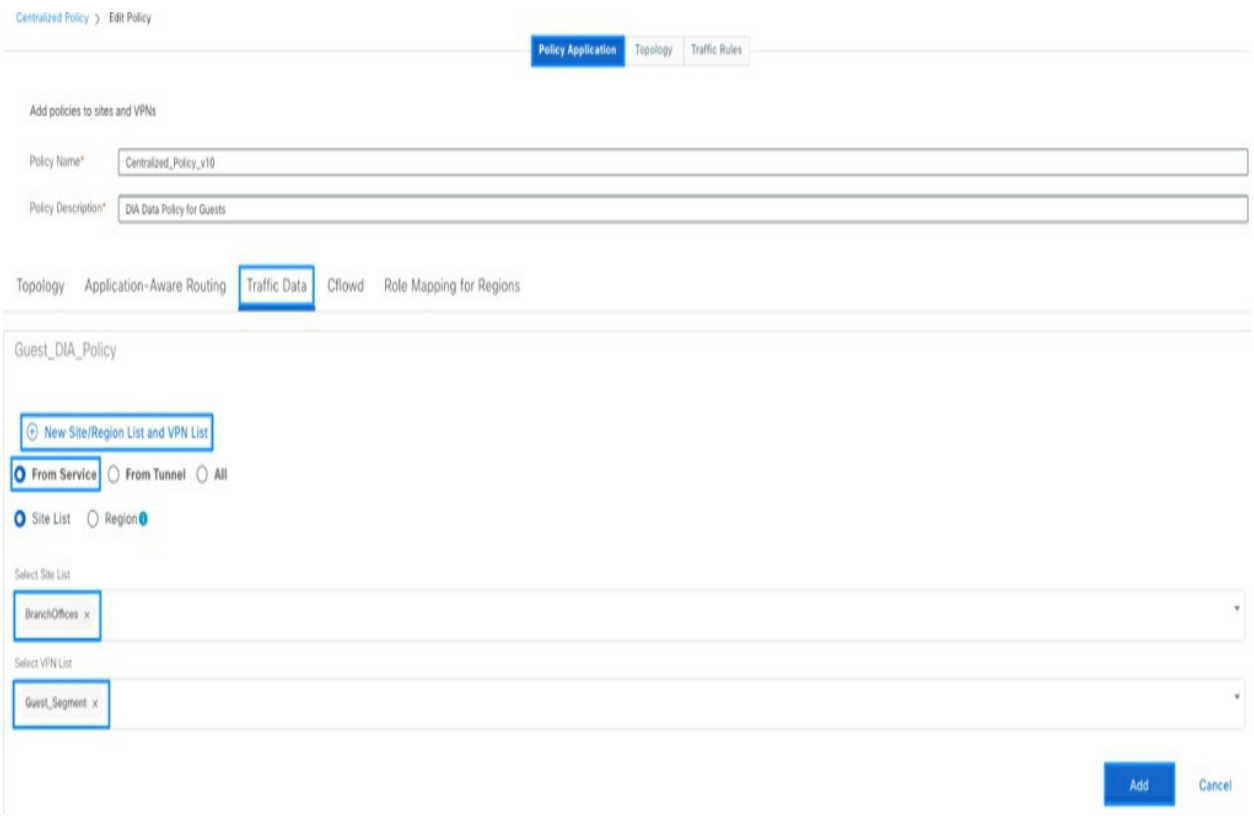


Figure 8-11 *Applying the New Data Policy*

Save these configurations by clicking the Add button. The policy should now be saved and activated on the SD-WAN fabric. Example 8-2 shows the new policy, highlighting relevant changes to the configuration from the full policy displayed in Example 7-24.

Technet24

Example 8-2 *Guest Internet Access Policy*

```
policy
 ! No changes were made to the control policies
 ! and they are omitted for brevity. The full c
 ! can be found in Example 7-24.
 !
 ! The newly created data policy is specified b
 ! the policy is applied to is specified in the
 ! apply-policy section at the end.
 !
data-policy _Guest_Segment_Guest_DIA_Policy
 vpn-list Guest_Segment
    sequence 1
     match
     destination-data-prefix-list BOGON_ADDR
    !
    action drop
     !
      ! The count action and counter are specif
      ! monitoring and troubleshooting.
      count GUEST_DROPPED_PKTS_2067976837
     !
    !
```

```
    ! In the second sequence, all other packets
    ! "nat use-vpn 0" syntax and also counted.
    ! of any source address were automatically


    sequence 11
      match
       source-ip 0.0.0.0/0
      !
      action accept
       nat use-vpn 0
       count GUEST_DIA_PKTS_2067976837
       set
        local-tloc-list
         color biz-internet
      !
    !
  default-action drop
!
lists
! Some lists without changes from Example 7-24
!
  data-prefix-list BOGON_ADDR
   ip-prefix 10.0.0.0/8
   ip-prefix 100.64.0.0/10
```

Technet24

```
   ip-prefix 127.0.0.0/8
   ip-prefix 172.16.0.0/12
   ip-prefix 192.168.0.0/16
  !
  site-list BranchOffices
   site-id 100-199
  !
  vpn-list Guest_Segment
   vpn 50
  !
!
apply-policy
 site-list Europe_Branches
  control-policy Euro_Reg_Mesh_with_FW_MultiTop
 !
 ! The newly created policy is applied to the S
 ! with the direction "from-service".
 !
 site-list BranchOffices
  data-policy _Guest_Segment_Guest_DIA_Policy f
  control-policy Branch_Extranet_Route_Leaking
  vpn-membership vpnMembership_1710051916
 !
 site-list DCs
  control-policy DC_Inbound_Control_Policy in
```

```
  !
  site-list North_America_Branches
   control-policy North_America_Reg_Mesh_with_FW
  !
 !
```

The structure of the data policy shown in Example 8-2 is very similar to the structure of the control policies discussed in Chapter 7. Each policy is a structured sequence of **match** statements that specify the criteria and a list of actions to take—that is, sequence types and sequence rules. In this policy, the first sequence rule is matching packets destined to the addresses in the BOGON_ADDR list. These packets are dropped and counted with the counter GUEST_DROPPED_PKTS. The second sequence matches all other traffic and forwards it out the VPN 0 interfaces to the Internet, using the **nat use-vpn 0** command. This traffic is also being tracked with the counter GUEST_DIA_PKTS. In this particular policy, the setting of the default action is irrelevant because all traffic that wasn't matched by sequence 1 will be matched by sequence 11.

Example 8-3 shows the effects of this policy: Users can still access the same resources on the public Internet, but traffic that should not be forwarded to the Internet is dropped instead of consuming that bandwidth.

Example 8-3 *Effects of the Data Policy on Users in the Guest VPN*

Technet24

```
! As the centralized data policy is enforced or
! is encoded as an OMP update by the SD-WAN cor
to the
! WAN-Edge. The policy is viewable with the "sh
!
BR2-Edge1# show sdwan policy from-vsmart
from-vsmart data-policy _Guest_Segment_Gues_DIA
 direction from-service
 vpn-list Guest_Segment
  sequence 1
   match
    destination-data-prefix-list BOGON_ADDR
   action drop
    count GUEST_DROPPED_PKTS_2067976837
  sequence 11
   match
    source-ip 0.0.0.0/0
   action accept
    count GUEST_DIA_PKTS_2067976837
    nat use-vpn 0
    no nat fallback
    set
     local-tloc-list
     color biz-internet
```

```
  default-action drop
from-vsmart lists vpn-list Guest_Segment
 vpn 50
from-vsmart lists data-prefix-list BOGON_ADDR
 ip-prefix 10.0.0.0/8
 ip-prefix 100.64.0.0/10
 ip-prefix 127.0.0.0/8
 ip-prefix 172.16.0.0/12
 ip-prefix 192.168.0.0/16
BR2-Edge1#
!
! The counters that are configured in the polic
policy
! data-policy-filter" command. Before any traf
!
BR2-Edge1# show sdwan policy data-policy-filter
data-policy-filter _Guest_Segment_Guest_DIA_Pol
 data-policy-vpnlist Guest_Segment
  data-policy-counter default_action_count
   packets 0
   bytes   0
  data-policy-counter GUEST_DIA_PKTS_206797683
   packets 0
   bytes   0
  data-policy-counter GUEST_DROPPED_PKTS_20679
```

Technet24

```
   packets 0
   bytes   0
!
! After sending four packets to 208.67.222.222
! has incremented to 5.
BR2-Edge1# ping vrf 50 208.67.222.222
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 208.67.222.22
!!!!!
Success rate is 100 percent (5/5), round-trip m
BR2-Edge1# show sdwan policy data-policy-filter
data-policy-filter _Guest_Segment_Guest_DIA_Pol
 data-policy-vpnlist Guest_Segment
  data-policy-counter default_action_count
   packets 0
   bytes   0
  data-policy-counter GUEST_DIA_PKTS_206797683
   packets 5
   bytes   500
  data-policy-counter GUEST_DROPPED_PKTS_20679
   packets 0
   bytes   0
! After sending five packets to 10.50.103.1, th
! incremented to 5.
!
```

```
BR2-Edge1# ping vrf 50 10.50.103.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.50.103.1,
.....
Success rate is 0 percent (0/5)
BR2-Edge1# show sdwan policy data-policy-filte
data-policy-filter _Guest_Segment_Guest_DIA_Po
 data-policy-vpnlist Guest_Segment
  data-policy-counter default_action_count
   packets 0
   bytes   0
  data-policy-counter GUEST_DIA_PKTS_206797683
   packets 5
   bytes   500
  data-policy-counter GUEST_DROPPED_PKTS_20679
   packets 5
   bytes   500
```



As shown in Example 8-3, the relevant portion of the data policy has been sent to the WAN Edge devices from the SD-WAN Controllers (formerly vSmarts). Note that this is different from the behavior for centralized control policies. Example 8-2 shows that there is a centralized control policy and a

Technet24

VPN membership policy that has also been applied to the site list BranchOffices; however, only the data policy is visible in the output of **show sdwan policy from-vsmart**. This is because the control policies are configured and executed on the SD-WAN Controllers directly and therefore have no reason to be advertised to the WAN Edge devices. The data policies, however, are configured and applied centrally on the SD-WAN Controller and therefore need to be advertised to the WAN Edge devices for enforcement at the data plane.

Network administrators can use the **show sdwan policy data-policy-filter** command to use counters to monitor how flows are being forwarded across the network (refer to Example 8-3). The first time the command is run, all the counters display zeros, as there have been no packets matched by the policy yet. After pinging a destination on the Internet with five packets, the GUEST_DIA_PKTS counter indicated that four packets were forwarded. Finally, the five pings sent to a bogon address are reflected in the GUEST_DROPPED_PKTS counter.

You can also see these effects by using the Simulate Flows tool, which can be found under the Troubleshooting menu on the Monitor > Devices > [*Device*] page discussed in Chapter 7. When simulating flows that are destined to 208.67.222.222, as shown in Figure 8-12, there are only a next-hop address and an interface specified for the flow. The output is lacking the remote system IP address, encapsulation, and color, which indicates that the flow is not going to be forwarded across the SD-WAN fabric but is instead going to

egress from the GigabitEthernet2 interface.

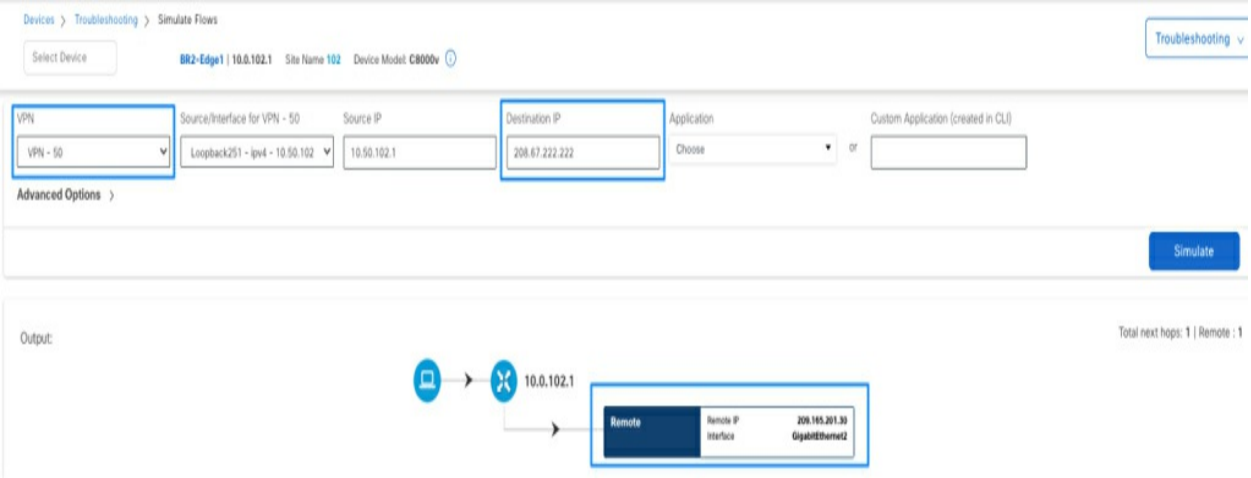


Figure 8-12 *Simulate Flows Output to 208.67.222.222 from VPN 50 Shows That Traffic Is Forwarded to the Internet*

On the other hand, when Simulate Flows is performed for traffic going to 10.50.103.1, it shows the unambiguous results displayed in Figure 8-13.

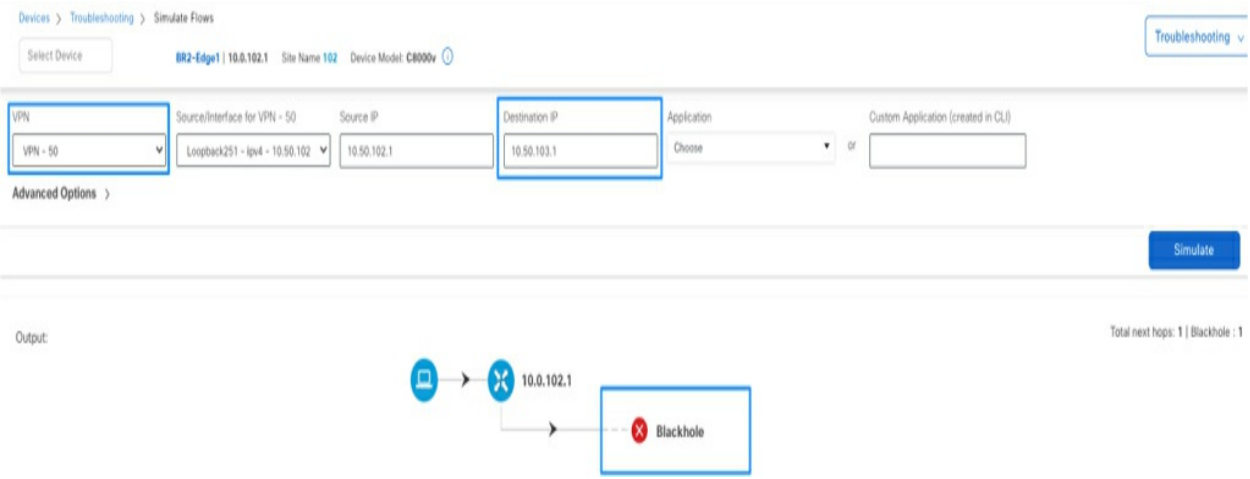


Figure 8-13 *Simulate Flows Output to 10.50.103.1 from VPN 50 Shows That Traffic Is Blackholed*

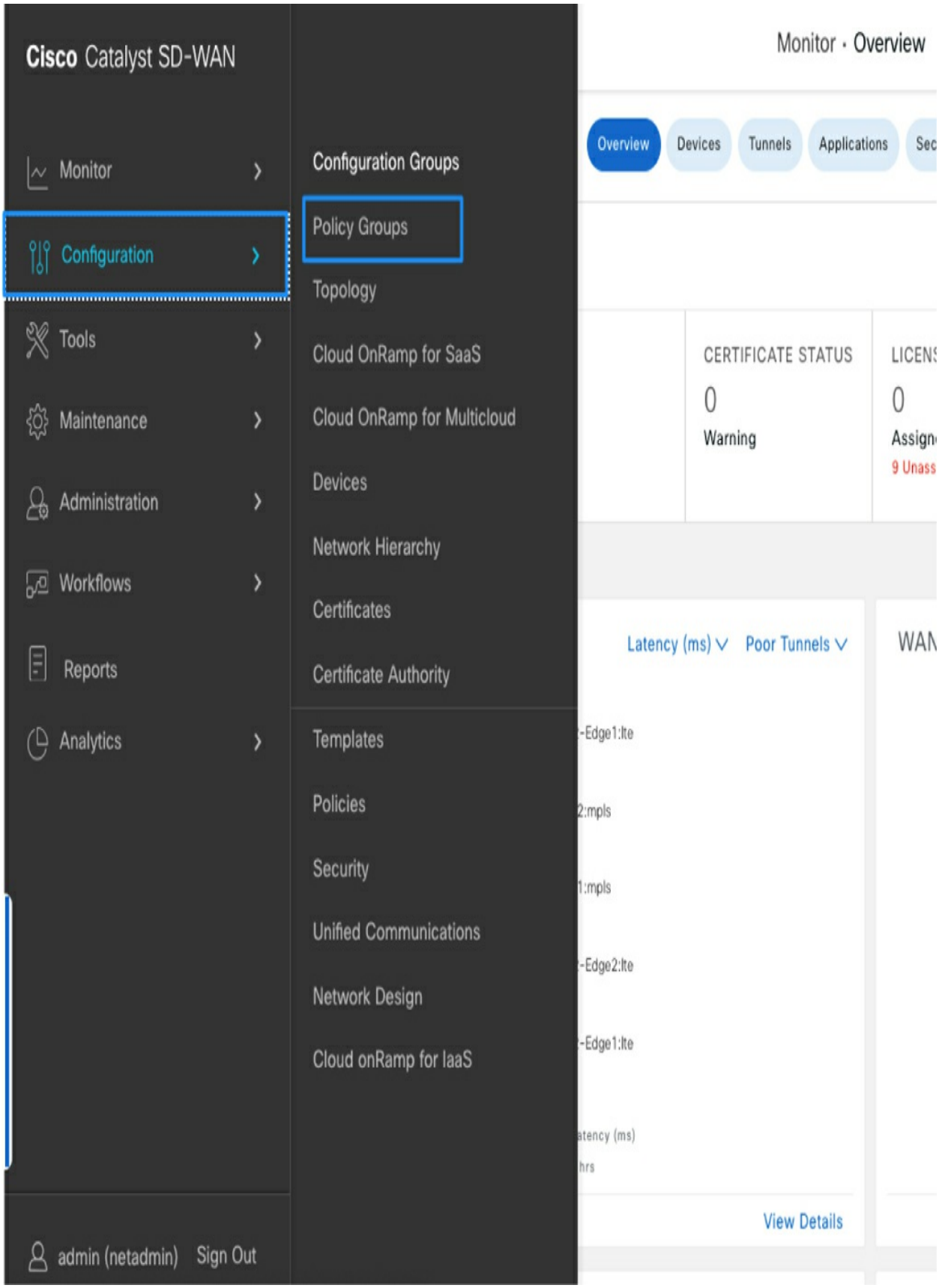
Direct Internet Access for the Guest VPN Using Policy Groups

Technet24

Starting in release 20.12, a simpler, reusable, and more structured approach for configuring policies in Catalyst SD-WAN Manager is available: policy groups. Chapter 7 covers topology groups as the equivalent to centralized control policies. The equivalent to centralized data policies is policy groups. Using policy groups involves using different policy profiles, such as application priority and SLA, embedded security, secure Internet gateway, and DNS security profiles. This use case shows how guest Internet access can be provided and assumes that VPN membership equivalent configuration is in place.

There are some prerequisites to using Policy Groups, such as ensuring the minimum software version for WAN Edge devices and configuring them to leverage configuration groups. For details, see the "Policy Groups Configuration Guide" On the Cisco website.

To begin working with policy groups, choose Configuration > Policy Groups, as shown in Figure 8-14.


Cisco Catalyst SD-WAN
Monitor
Configuration
Tools
Maintenance
Administration
Workflows
Reports
Analytics
admin (netadmin)
Sign Out
Configuration Groups
Policy Groups
Topology
Cloud OnRamp for SaaS
Cloud OnRamp for Multicloud
Devices
Network Hierarchy
Certificates
Certificate Authority
Templates
Policies
Security
Unified Communications
Network Design
Cloud onRamp for IaaS
Monitor · Overview
Overview
Devices
Tunnels
Applications
CERTIFICATE STATUS
0
Warning
0
Latency (ms)
Poor Tunnels
View Details

Technet24

Figure 8-14 *Opening the Policy Groups Screen*

At this point, an Application Priority & SLA policy profile is selected, as shown in Figure 8-15. When you select Add Application Priority Policy, a window will pop up requesting a policy name and description, as shown in Figure 8-16.

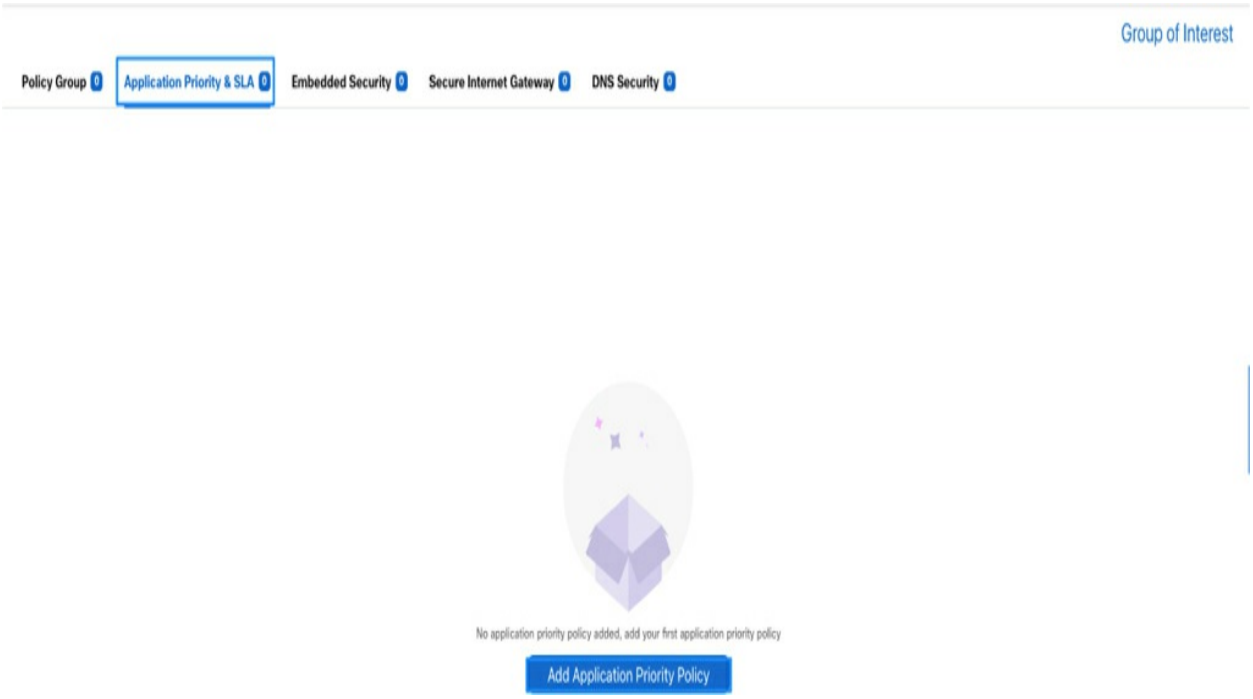


Figure 8-15 *Creating an Application Priority Policy*

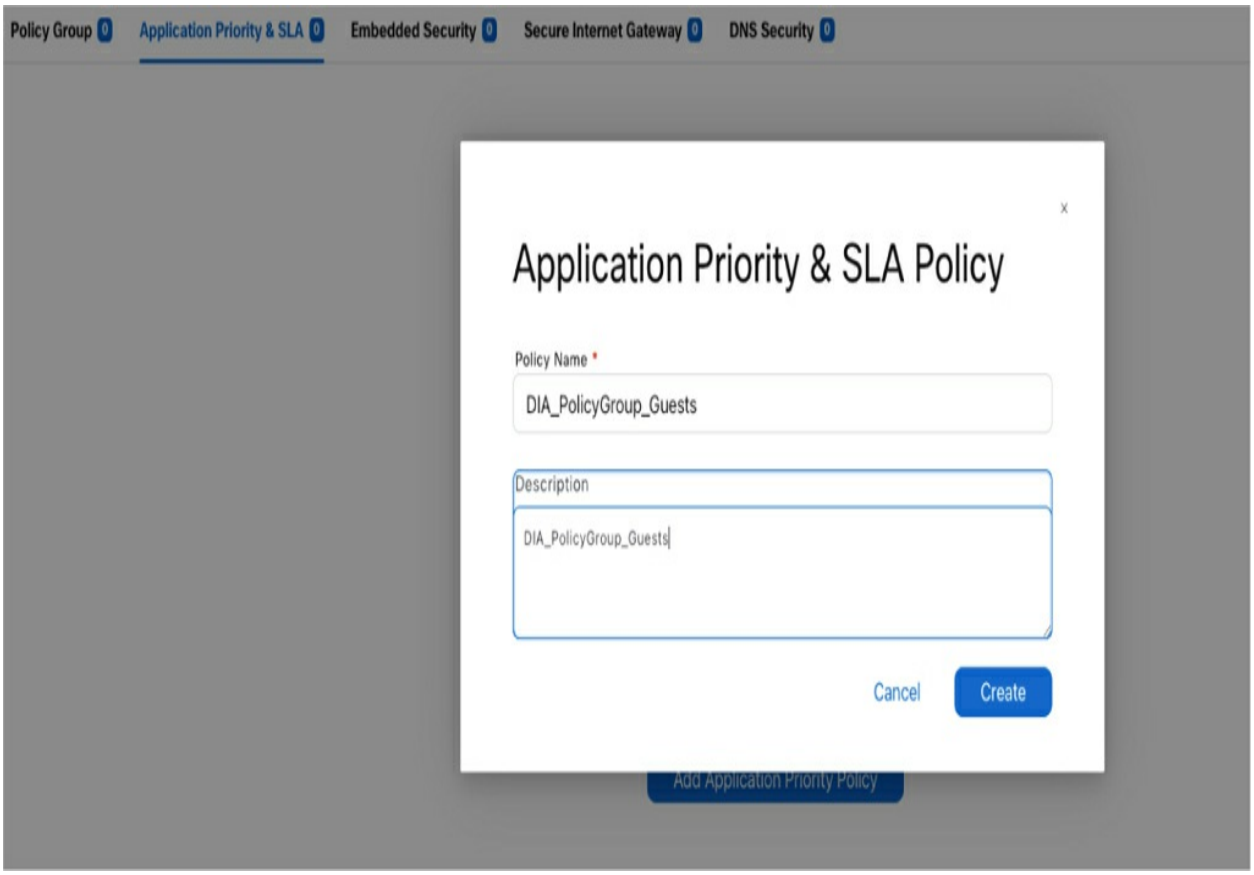


Figure 8-16 *Providing a Name and Description for the Newly Created Policy*

In the Application Priority & SLA Policy window, you can see that smart defaults are provided to make it easier to create policies. Click Create. To proceed, select Advanced Layout, as shown in Figure 8-17. This allows you to use a blank canvas, as in the previous data policy model.

Technet24

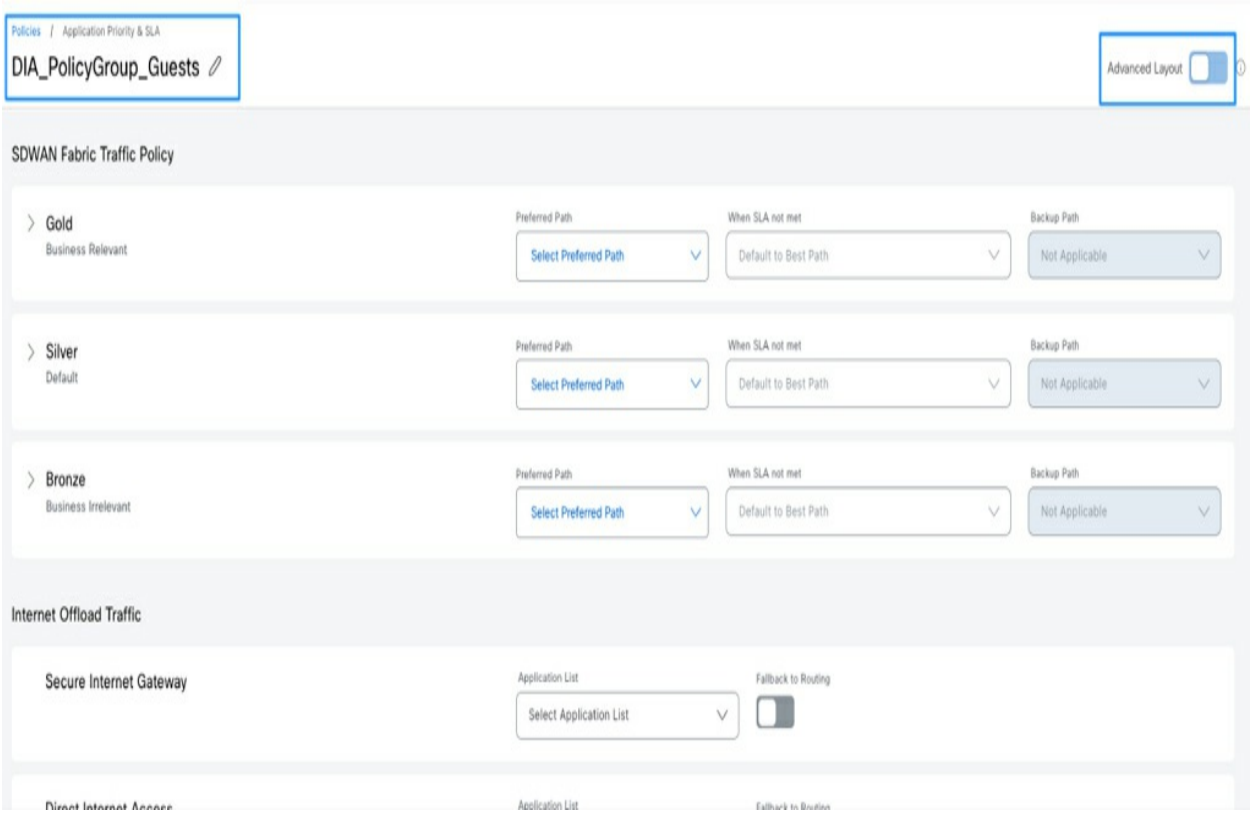


Figure 8-17 *Selecting Advanced Layout*

Click Add Traffic Policy, and the Add Traffic Policy List window appears. Fill in the name, select VPN 50—Guest Users, and select Service Direction, as shown in Figure 8-18, and then click Add.

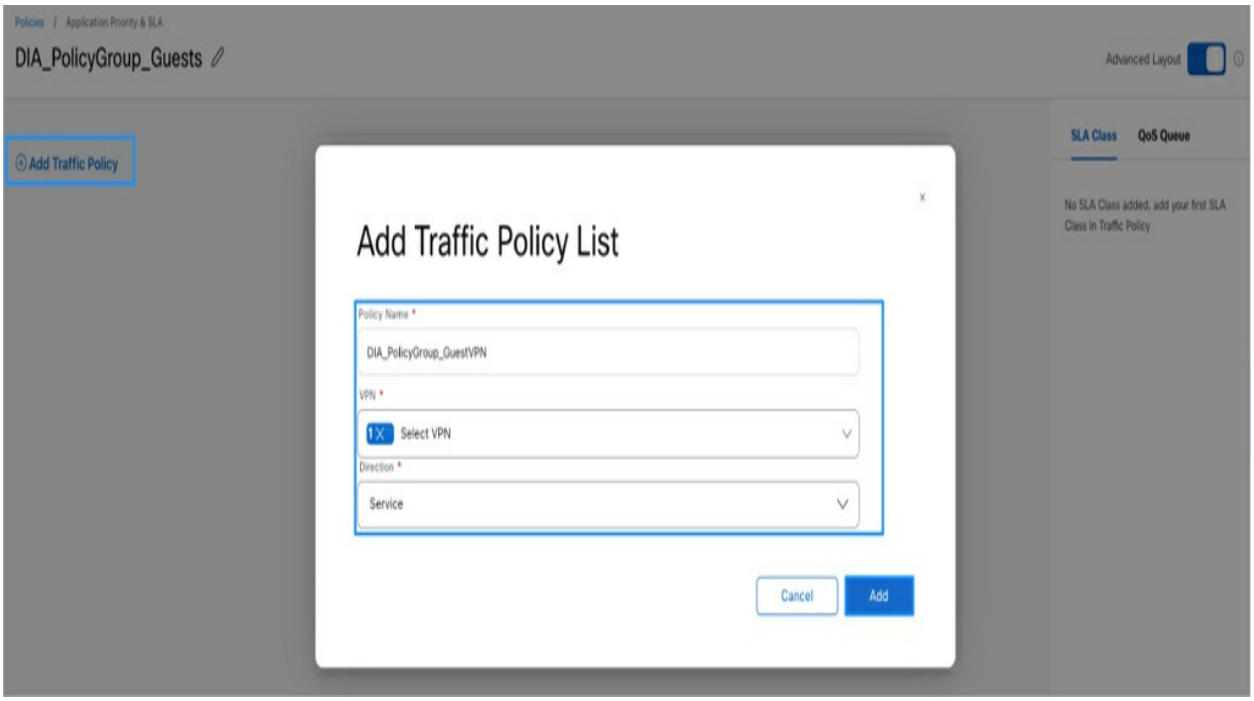


Figure 8-18 *Add Traffic Policy List Window*

Inside the recently created policy, you need to click Add Rules and provide a name for the sequence (in this case, BOGON_ADDR). You also need to select Add Match and then select Destination and (optionally) enter a value or create an object, as shown in Figure 8-19.

Technet24

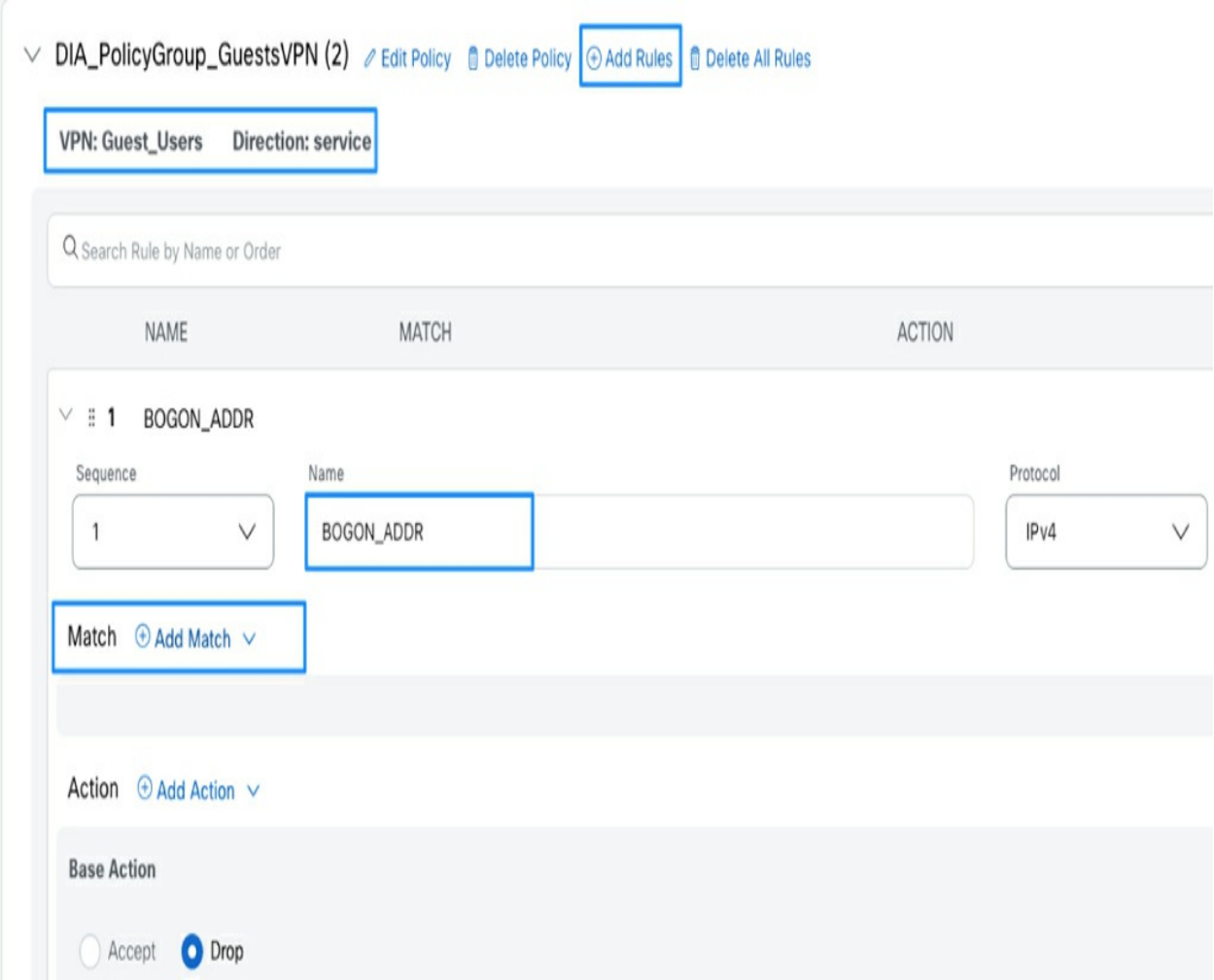


Figure 8-19 *Advanced Layout BOGON Sequence Rule*

Select Create New for Destination Data Prefix (Ipv4), and a Data Prefix List window pops up. Enter a name for the list and prefix values, separated by commas (see Figure 8-20). Then click Save.

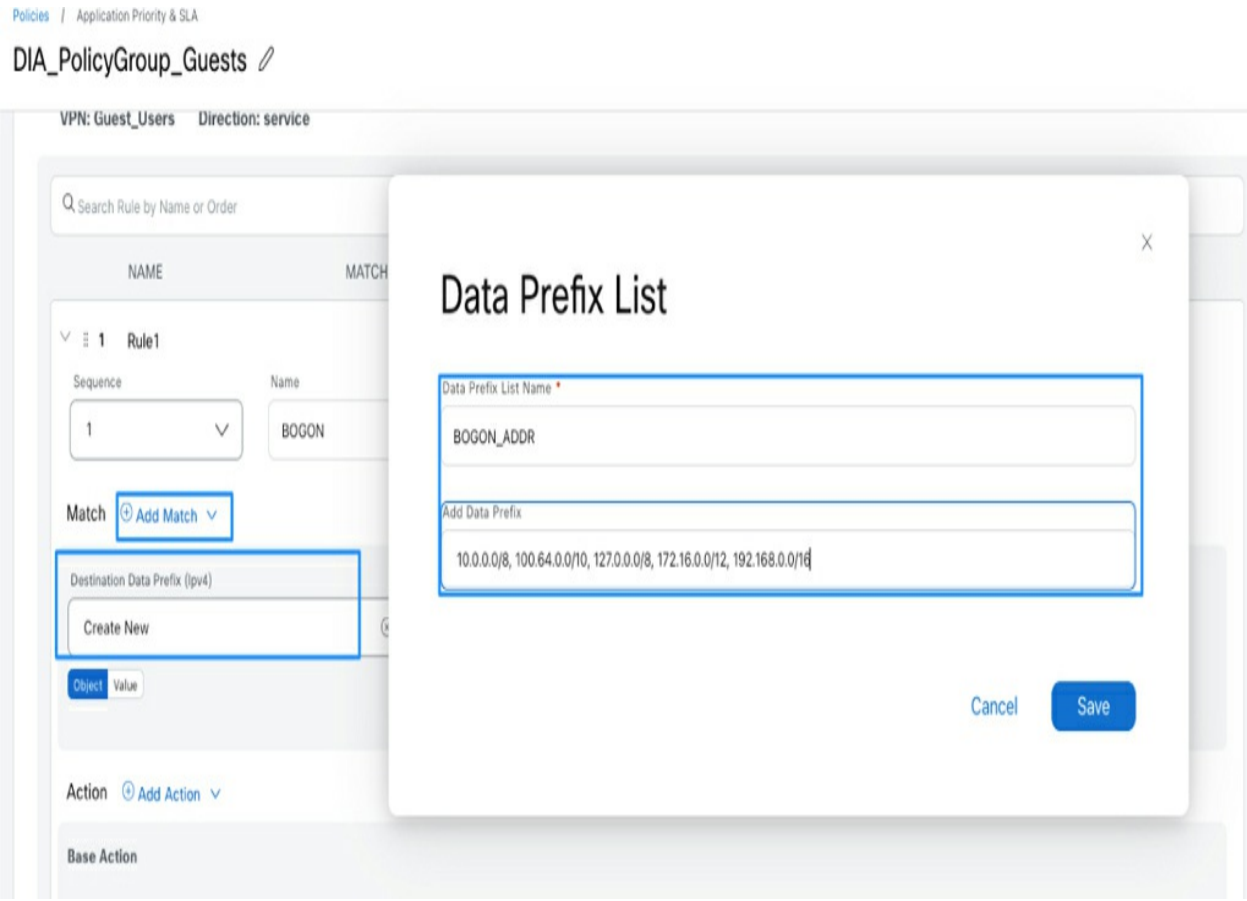


Figure 8-20 *Creating a Data Prefix List Inside a Policy Group Profile*

Once the data prefix list is created, select it from the Destination Data Prefix (Ipv4) drop-down list to use it as a **match** statement. Figure 8-21 shows the selections you need to make for the **match** statement as well as for the base action (Drop) and the counter name (GUEST_DROPPED_PKTS). Click Save Match and Actions.

Technet24

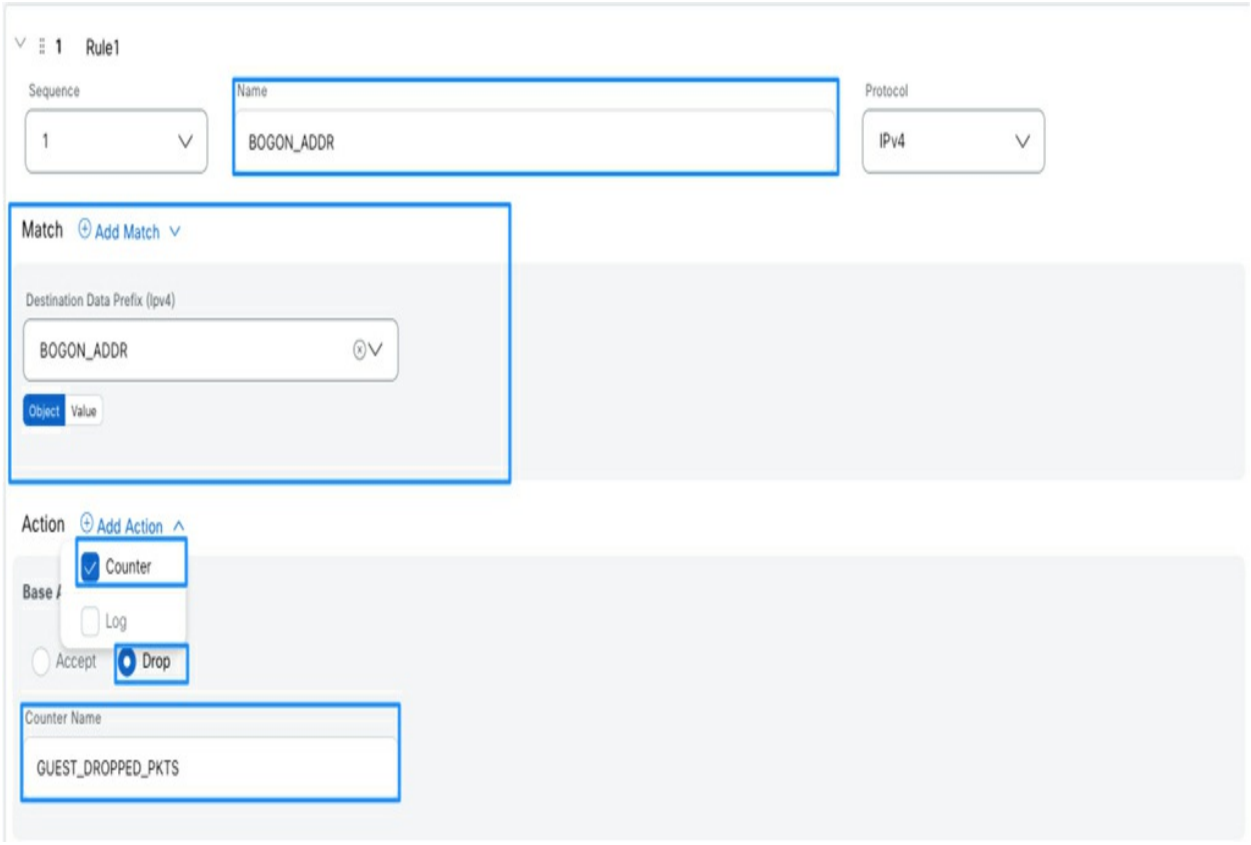


Figure 8-21 *Completing the Match and Actions for the BOGON Sequence*

Because direct Internet access is needed, you need to add a second rule to the policy. Figure 8-22 shows the selections you need to make. Leave Match empty in order to match any and set Base Action to Accept. Click +Add Action and configure Local TLOC by selecting biz-internet color along with IPsec encapsulation and VPN ID 0 as the NAT VPN. Then click Save Match and Actions.

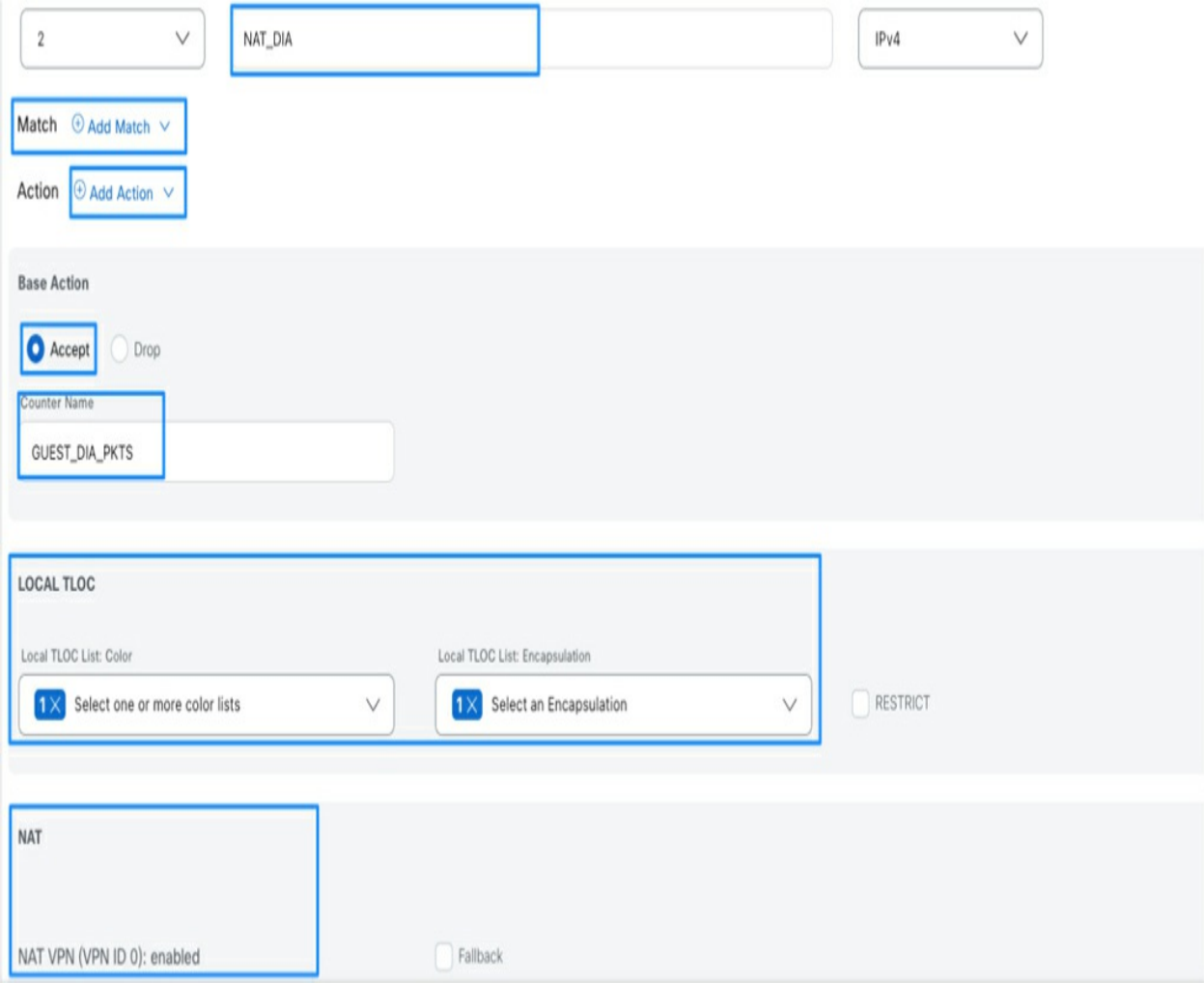


Figure 8-22 *Completing the Match and Actions for the NAT VPN*

The result is two sequence rules in the Policy Group profile, as shown in Figure 8-23. Make sure the rules are correct and click Save.

Technet24

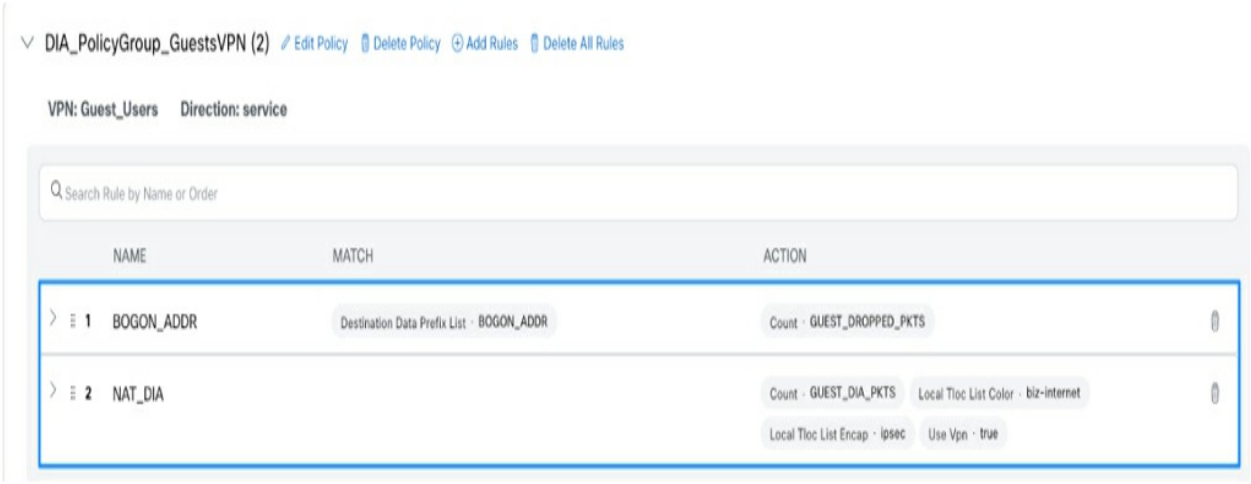


Figure 8-23 *Policy Profile Result*

Once the Application Priority & SLA profile is created, it is associated to a Policy Group, which will group different sets of policy profiles. Figure 8-24 shows how to create a policy group by providing a name and a description. Then click Create.

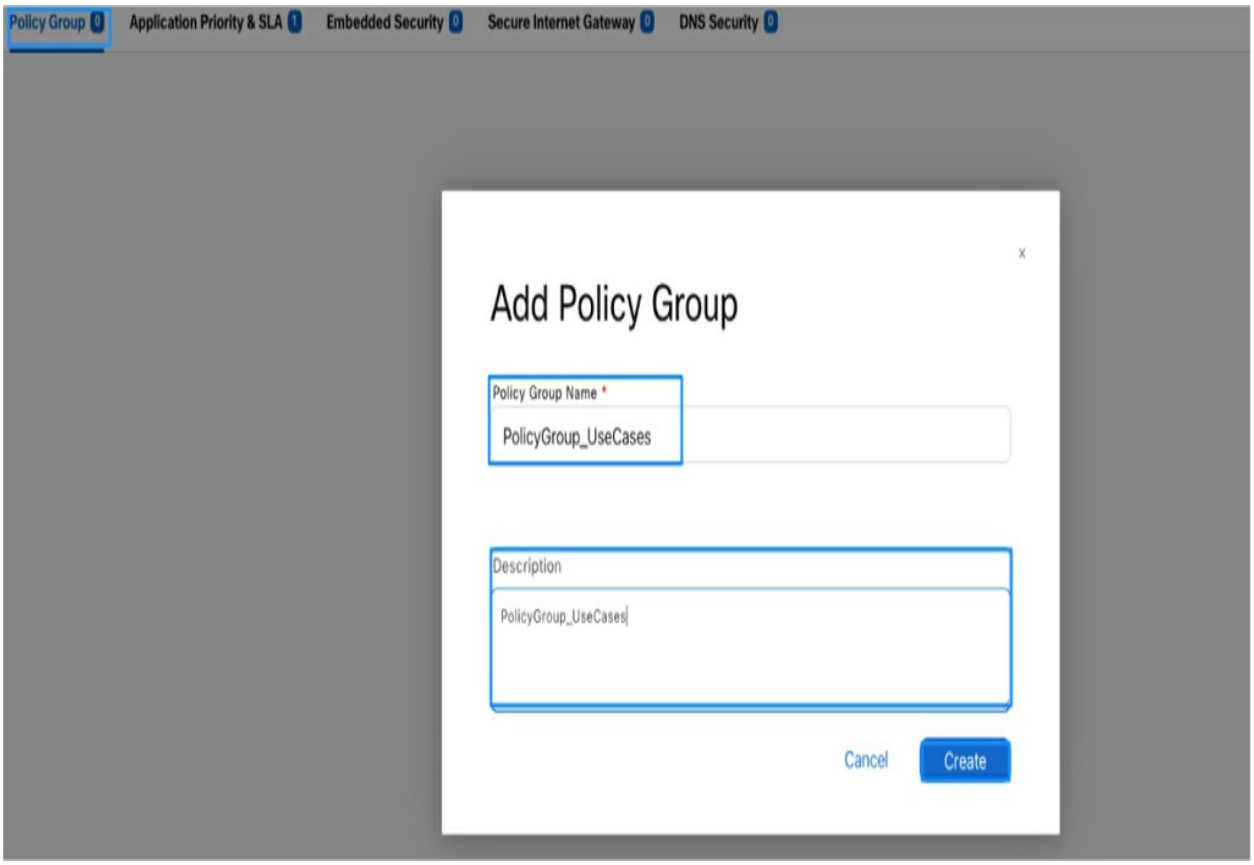


Figure 8-24 *Creating a Policy Group*

In the Policy Group screen, as shown in Figure 8-25, you can select the drop-downs for the policy profile options. In this case, you associate the newly created application priority profile DIA_PolicyGroup_Guests. An intermediate step is required to associate this policy group to devices; only devices associated with a Configuration Group are available. Once the desired devices are associated, click Deploy.

Technet24

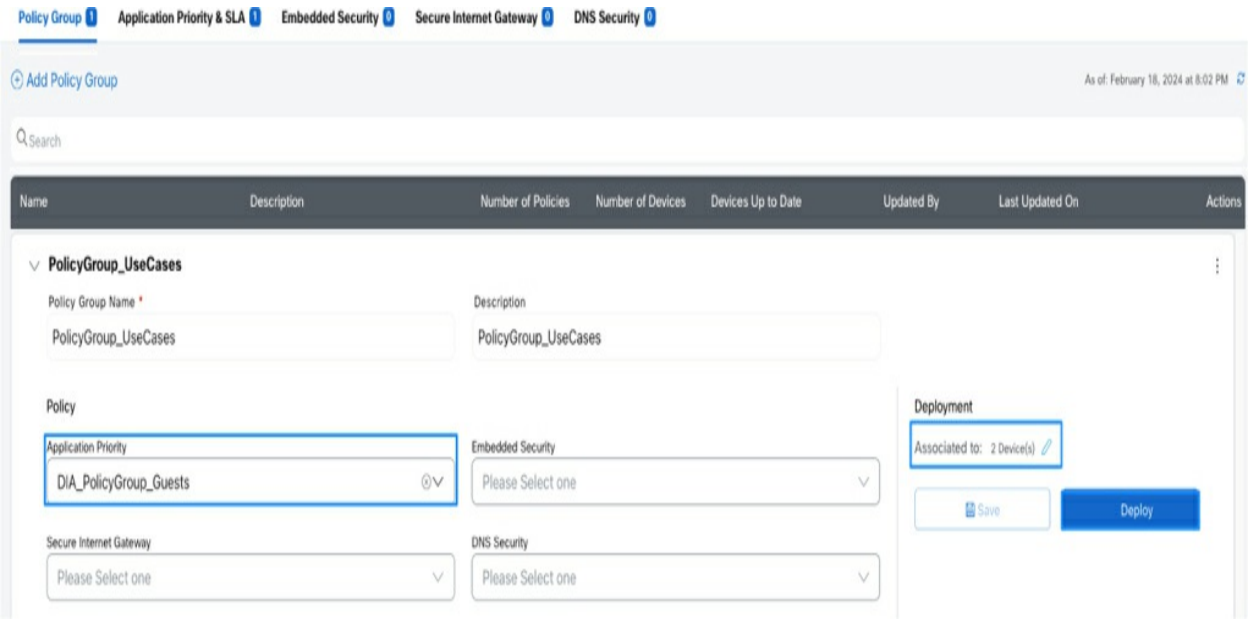


Figure 8-25 *Associating the Application Priority with the Policy Group and the Device*

Use Case 10 Review

This use case shows how to create a simple data policy and Application Priority & SLA profile and applied them to VPN 50. In this example, you have seen how data policies and policy groups can be used to manipulate the forwarding path of traffic. The two simple examples for the guest user VPN illustrate that the structure of this type of policy, consisting of **match** and **action** sequence rules, is very similar to the structure of control policies (refer to Chapter 7). While the purpose of this policy—dropping packets destined to a specific destination—is relatively straightforward, it is easy to see how other policies could be used for much more intricate tasks. For example, it would be trivial to drop traffic destined to a specific port while permitting traffic to other ports with a data policy—and it would be impossible to do so

with a control policy.

Use Case 11: Direct Cloud Access for Trusted Applications

In addition to providing Internet access for guest users, many organizations leverage local Internet breakout for employees as well. However, there are significant security implications involved with permitting direct Internet access from every branch office to the entirety of the Internet. (You can find detailed discussions of the security features that are built in to the WAN Edge routers in Chapter 11, "Cisco Catalyst SD-WAN Security.") One of the ways that organizations can choose to limit the security implications of these choices is to restrict direct access to the Internet from the branch to certain applications or destinations that are deemed to be trusted or lower risk. Such applications may include enterprise services such as Microsoft 365, Google apps, and Salesforce.com. The terms *direct cloud access* and *direct Internet access* are often used interchangeably throughout this chapter to refer to policies that selectively permit access to destinations on the Internet.

In this use case, you will see how to build a data policy that allows users to access a specific trusted application, Cisco Webex, directly from the local branch, while still requiring all other Internet traffic to traverse the main security perimeter through the data centers. Figure 8-26 illustrates this traffic pattern for corporate users.

Technet24

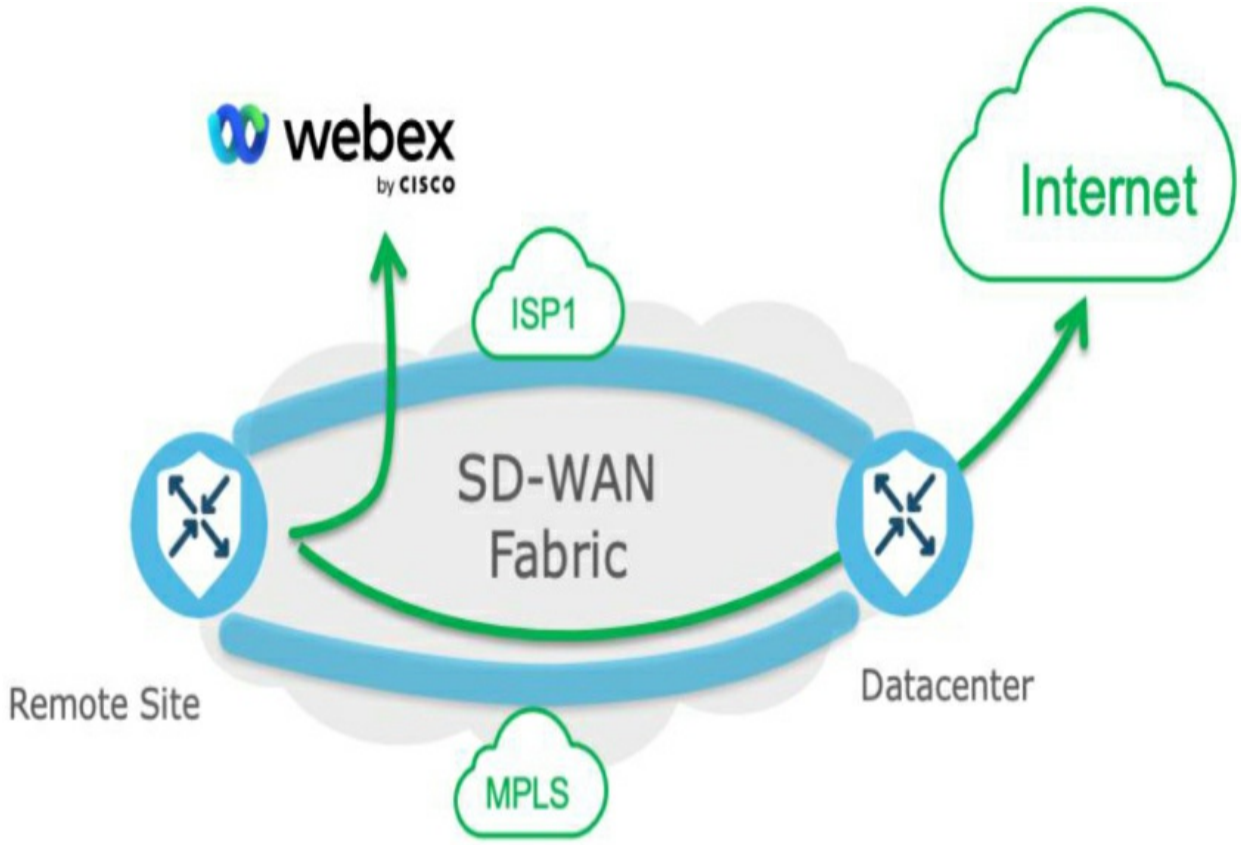


Figure 8-26 *Desired Forwarding Paths Are Different for Each Application*

Before the policy is applied, all the traffic from the corporate users behind BR2-Edge1 traverse through DC1 to reach the public Internet. You can see this with Simulate Flows tool results in Figure 8-27. The remote system IP address of the device to which the traffic is going to be forwarded is listed as 10.0.10.1 in this output, and the application webex-meeting is specified in the flow criteria.

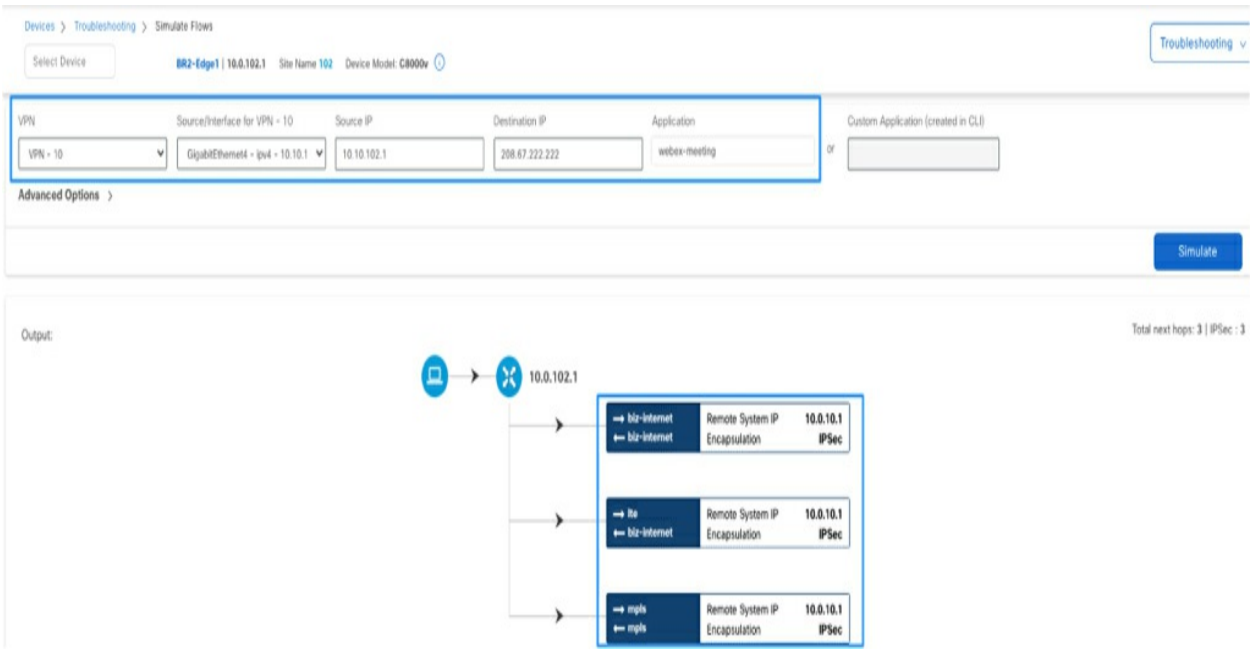


Figure 8-27 *Webex Traffic from Users in VPN 10 Would Be Forwarded to DC1 WAN Edge Routers*

Note

In Figure 8-27 and Figure 8-28, different applications are specified, and an Internet-routable address is used as the destination (a Cisco Umbrella DNS address). By specifying a combination of different applications, DSCP markings, addresses, ports, and/or protocols, the Simulate Flows tool is a particularly powerful tool that helps network administrators understand how traffic is flowing through the network.

In Figure 8-27 and Figure 8-28, both Webex traffic and YouTube traffic are forwarded across the SD-WAN fabric to DC1-Edge1 or any site that is

Technet24

advertising a default route, as indicated by the remote system IP address 10.0.10.1. These application flows would be load-shared across the MPLS, LTE, and BIZ-INTERNET paths, as indicated by the multiple paths in the output. These outputs, taken before the updated policy is applied, show that the forwarding path for the two applications is the same.

Note

As discussed in Chapter 7, the best-path selection algorithm picks the best route to a destination and advertises it through Overlay Management Protocol (OMP) to WAN Edge devices. For this case, the only available path is through remote system IP 10.0.0.1 because the control policy to influence symmetry and prefer DC1-Edge1 (preference 500) is still in place. Under normal conditions, all available paths would be present in the output. Remember that settings such as ecmp-limit or send-path-limit also influence the number of paths WAN Edge devices will use to forward traffic to its destination. For simplicity in this use case, only DC1 is advertising a default route.

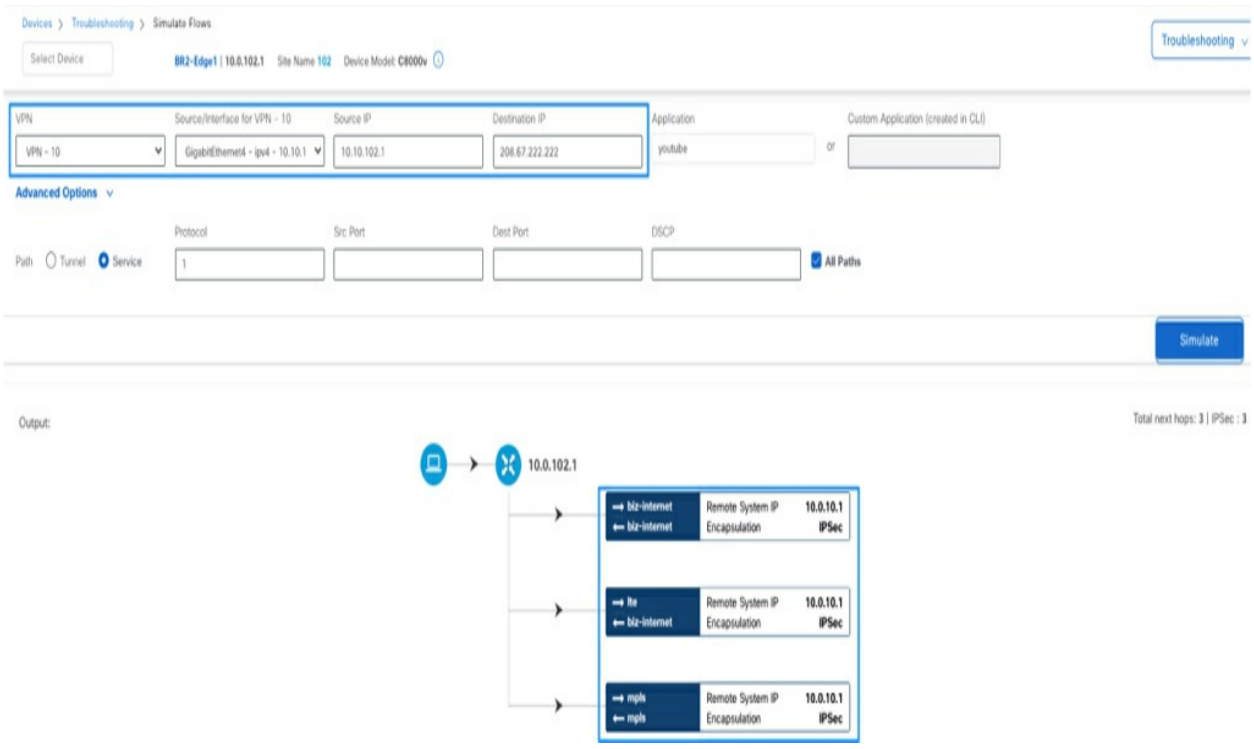


Figure 8-28 *YouTube Traffic from Users in VPN 10 Would Be Forwarded to DC1 WAN Edge Routers*

In order to meet the requirements of this use case and forward the Webex traffic directly out to the Internet while continuing to backhaul all other traffic, including YouTube traffic, across to the data centers, the centralized policy from Example 8-2 is modified to include a new data policy that matches the Webex application and forwards it out the local interface, as shown in Example 8-4.

Note

SD-WAN Application Intelligence Engine (SAIE) provides the ability to look into a packet beyond the basic header information. It relies on the Network-Based Application

Technet24

Recognition (NBAR) and Software Defined-Application Visibility Control (SD-AVC) engines to achieve tasks like application recognition and classification based on the characteristics of the traffic flow to set an action to it through policies. This setting configuration is explained in Chapter 10, "Localized Policies."

Example 8-4 *A Corporate Direct Cloud Access Policy*

```
policy
 ! No changes were made to the control policies
 ! and they are omitted for brevity. The full c
 ! can be found in Example 7-24.
 !
 ! The data policy below specifies two differen
 ! individual set of sequences with different m
 ! is applied to the site list.
 !
 data-policy
_Corporate_Segment_Corporate_DCA_Guest_Segment_
  vpn-list Corporate_Segment
    sequence 1
     ! In the new sequence for the CorporateVPN
```

```
     ! for the TRUSTED_APPS. The action (nat us
was
     ! used for guest internet access. A new co
     ! for monitoring.
    match
     app-list TRUSTED_APPS
      source-ip 0.0.0.0/0
     !
      ! The nat fallback configuration specifie
event
      ! that the NAT enabled interface, biz-int
     action accept
      nat use-vpn 0
       nat fallback
      !
      count CORP_DCA_-1626632101
     !
    !
   ! All non-webex traffic will be matched by t
as
   ! normal across the fabric.
   !
  default-action accept
 !
 ! The Guest_DIA_Policy that was configured as
```

Technet24

```
!
 vpn-list Guest_Segment
  sequence 1
    match
     destination-data-prefix-list BOGON_ADDR
    !
    action drop
     count GUEST_DROPPED_PKTS_-1626632101
    !
   !
   sequence 11
    match
     source-ip 0.0.0.0/0
    !
    action accept
     nat use-vpn 0

     count GUEST_DIA_PKTS_-1626632101
     set
       local-tloc-list
       color biz-internet
    !
   !
 default-action drop
!
```

```
 lists
 ! <<<Some lists without changes are omitted fo
 !
  app-list TRUSTED_APPS
   app webex
   app webex-app-sharing
   app webex-meeting
   app webex_weboffice
   app webex-audio
   app webex-media
   app webex-control
  !
  vpn-list Corporate_Segment
   vpn 10
  !
  vpn-list Guest_Segment
   vpn 50
  !
 !
!
apply-policy
 site-list Europe_Branches
  control-policy Euro_Reg_Mesh_with_FW_MultiTop
 !
site-list BranchOffices
```

Technet24

```
 data-policy _Corporate_Segment_Corporate_DCA_C
from-service
 vpn-membership vpnMembership_1718052183
 !
 site-list DCs
  control-policy DC_Inbound_Control_Policy in
 !
 site-list North_America_Branches
  control-policy North_America_Reg_Mesh_with_FW
 !
!
```



Example 8-4 highlights the modifications to the data policy that were made in order to implement the direct cloud access functionality for corporate users. This policy builds on the policy that was created in Use Case 10 and adds support for corporate users in a second VPN (VPN 10) with an entirely different set of rules. It is important to understand that these two sets of sequences for the two different VPNs have been concatenated into a single data policy when activated on the SD-WAN Controller (formerly vSmart), even though they are administered in SD-WAN Manager (formerly vManage) as two separate data policies (as shown in Figure 8-29). In this case, the policy

_Corporate_Segment_Corporate_DCA_Guest_Segment_Guest_DIA_Policy affects the VPNs specified by both the Corporate_Segment and Guest_Segment lists, albeit with different policies specified in their respective sequence sets. Furthermore, there is no reference to the VPNs that the data policy is applied to in the apply-policy stanza at the end of the centralized policy; this is configured by the VPN lists referenced in the data policies.

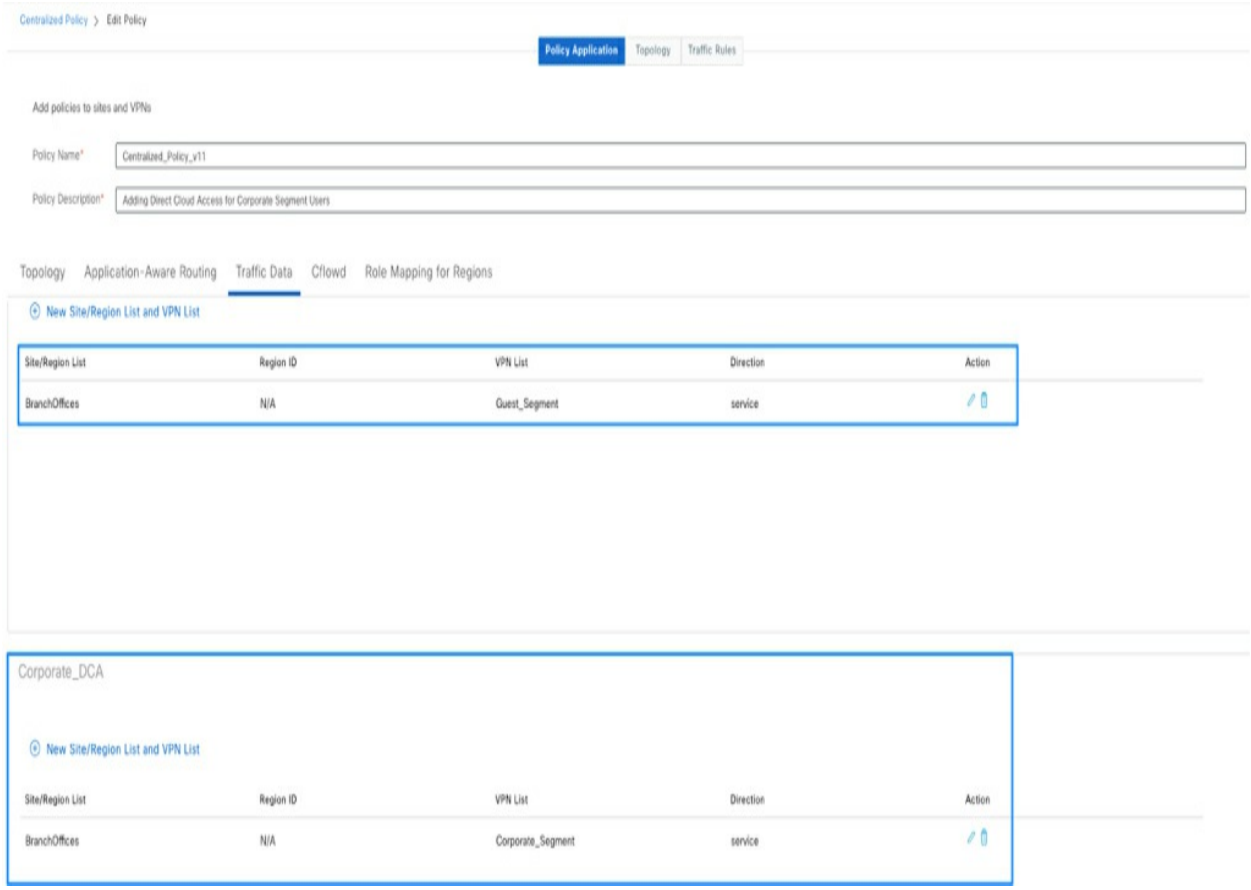


Figure 8-29 *Applying a Policy Configuration Showing That Corporate and Guest Data Policies Are Separate Policies*

Once the policy is activated, the results can be validated using the Simulate

Technet24

Flows tool. As shown in Figure 8-30, Webex traffic now egresses directly to the Internet. This flow pattern is different from what was observed for Webex traffic before the policy was applied, in Figure 8-27, and it's similar to what was seen for traffic in the Guest VPN in Figure 8-12.

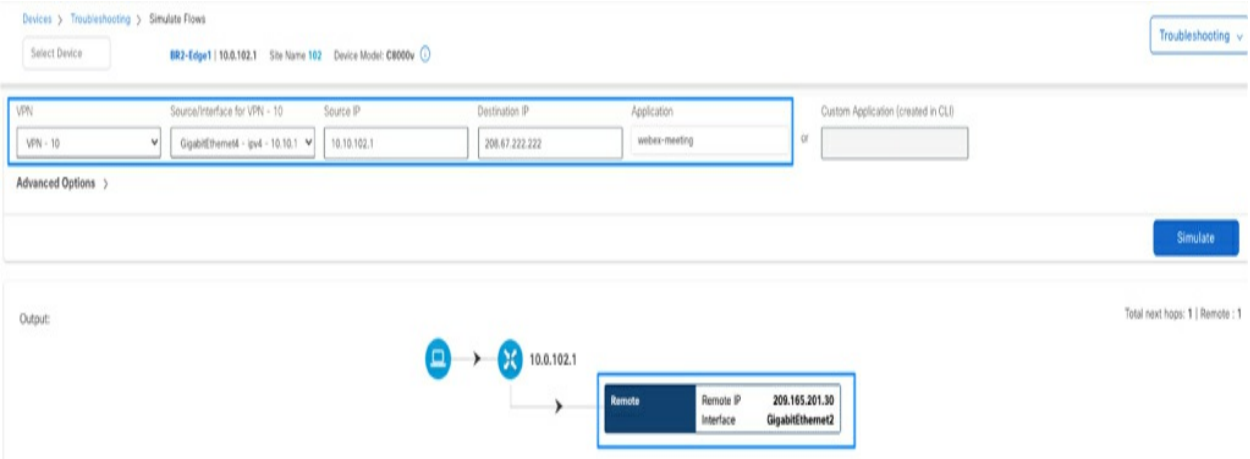


Figure 8-30 *Webex Traffic from Users in VPN 10 Would Egress Directly to the Internet*

Conversely, the forwarding pattern for YouTube traffic, as shown in Figure 8-31 after the policy is activated, is the same as in Figure 8-28 before the policy was activated. All the YouTube traffic is going to be forwarded across the fabric to the data center and will be inspected by the traditional security perimeter there.

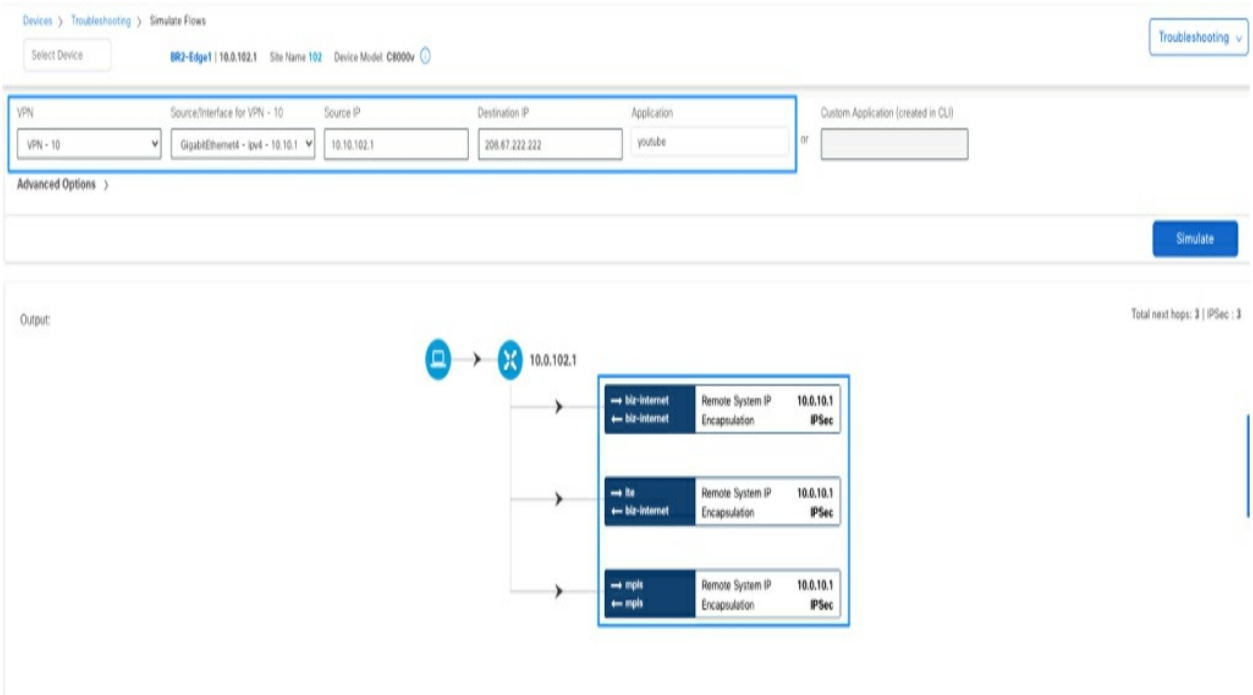


Figure 8-31 *YouTube Traffic from Users in VPN 10 Would Continue to Be Forwarded to the DC1 WAN Edge Routers*

As highlighted in Figure 8-7 and Example 8-4, the **nat use-vpn 0** action has the optional configuration argument **nat fallback**. This argument allows the administrator to specify the desired behavior in the event of a failure. With **nat fallback** enabled, in the event that all the NAT-enabled VPN 0 interfaces on the local edge were in a non-operational state, the traffic would follow the forwarding path determined by the routing table. This would typically mean that the traffic traverses the fabric and egresses at a different site (such as a data center). Figure 8-32 illustrates this traffic pattern.

Technet24

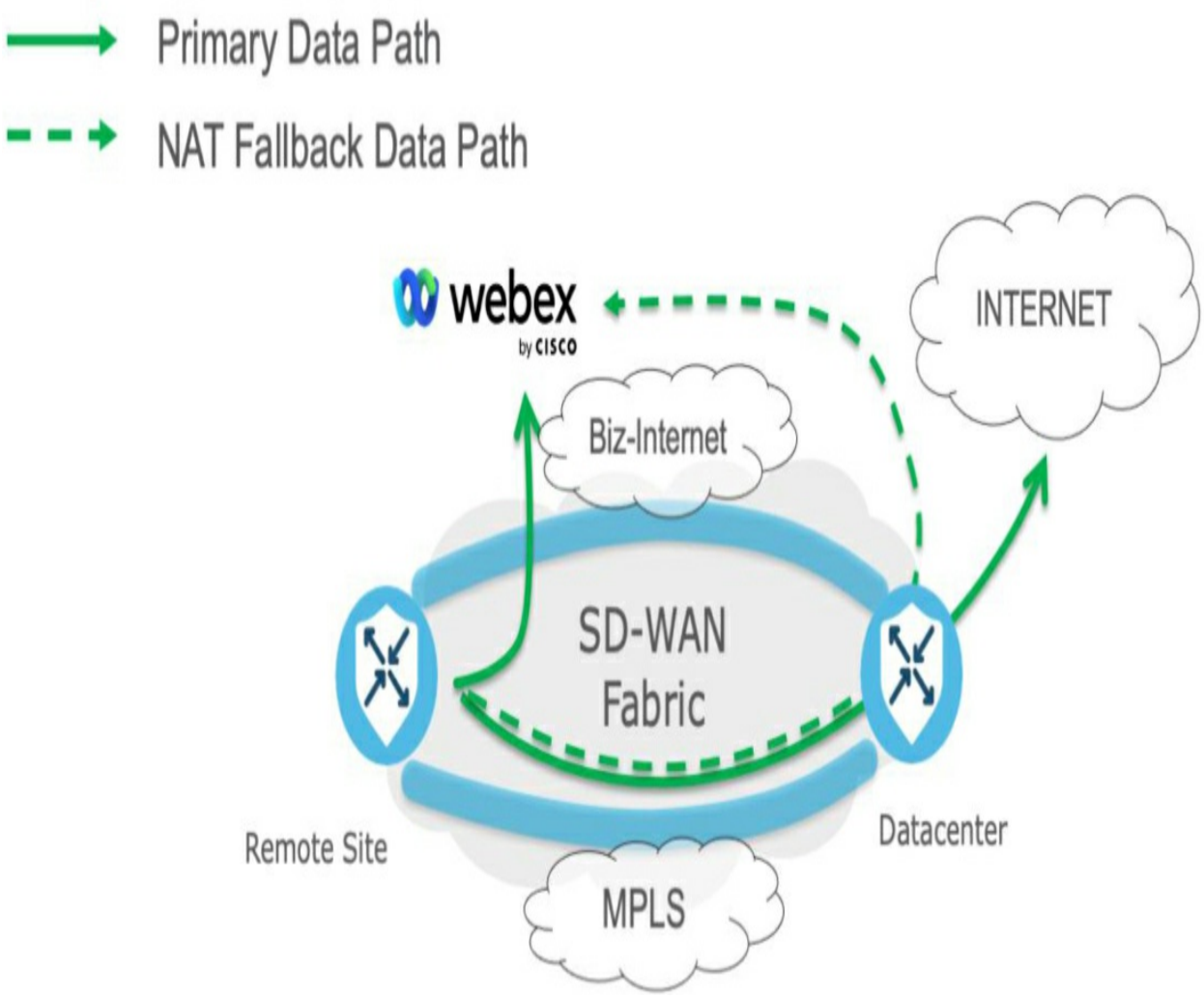


Figure 8-32 *Illustrated* **nat fallback** *Data Path*

Note

A non-operational state for a NAT-enabled interface can occur in two main ways: a hard failure, where the interface goes down completely, or a soft failure, where a NAT DIA tracker associated with the interface changes status to down.

In Use Case 10, the **nat use-vpn 0** action is configured for guest Internet

access traffic without **nat fallback** enabled. If all of the local interfaces with NAT configured were to go down, guest traffic would not fall back and would be forwarded across the SD-WAN fabric and would instead be blackholed. In this use case, the data policy for the corporate VPN does have the **nat fallback** option configured, and this allows for the Webex traffic to be backhauled through the data center in the event of a failure of the local Internet connection. Example 8-5 shows both traffic patterns.

Example 8-5 also introduces a new tool for monitoring the forwarding decision of flows from the CLI: **show sdwan policy service-path**. This **show** command is the CLI equivalent of the Simulate Flows tool that has been shown inside the SD-WAN Manager GUI.

Note

For direct Internet access forwarding scenarios, NAT Direct Internet Access tracking is used to determine if the Internet or external network is unavailable and hence the Internet-bound traffic needs to be routed through an alternate path.

Example 8-5 *Webex Traffic with Failure of the Local Internet Connection*

```
! When all of the transport interfaces as well
!
```

Technet24

```
BR2-Edge1# show ip interfaces brief
Interface              IP-Address      OK? Metho
GigabitEthernet1       192.168.20.6    YES othe
GigabitEthernet2       209.165.201.6   YES othe

BR2-Edge1#sh endpoint-tracker interface  Gigabi
Interface                    Record Name
Family  RTT in msecs    Probe ID   Next Hop
GigabitEthernet2           Tracker_Biz_Internet
9             1        209.165.201.30
!
! With all of the transports up / up, Webex tra
! locally. This output matches Figure 8-30.
!
BR2-Edge1# show sdwan policy service-path vpn 1
source-ip 10.10.102.1 dest-ip 208.67.222.222 pr
Number of possible next hops: 1
Next Hop: Remote
  Remote IP: 209.165.201.30, Interface GigabitE
BR2-Edge1#
!
! VPN 50 has connectivity to resources on the i
!
BR2-Edge1# ping vrf 50 208.67.222.222
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 208.67.222.22
!!!!!
Success rate is 100 percent (5/5), round-trip r
BR2-Edge1#
!
! Turning GigabitEthernet2 to a down state (har
of the local internet link:
!
BR2-Edge1#show ip interface brief
Interface              IP-Address      OK? Metho
GigabitEthernet1       192.168.20.6    YES othe
GigabitEthernet2       209.165.201.6   YES othe
<<<Output omitted for brevity>>>
BR2-Edge1#
BR2-Edge1#sh endpoint-tracker
Interface                    Record Name
Family   RTT in msecs    Probe ID   Next Hop
GigabitEthernet2                Tracker_Biz_In
25            0           209.165.201.30
!
! With the local internet connection down, Webe
! the fabric to destinations 209.165.201.1 and
table,
! we can see that this is DC1-Edge1 (System IP
BR2-Edge1# show sdwan policy service-path vpn 1
```

Technet24

```
source-ip 10.10.102.1 dest-ip 208.67.222.222 pr
Number of possible next hops: 2
Next Hop: IPsec
  Source: 209.165.202.130 12346 Destination: 20
Remote Color: biz-internet Remote System IP: 10
Next Hop: IPsec
  Source: 192.168.20.6 12346 Destination: 192.1
Remote Color: mpls Remote System IP: 10.0.10.1


!
! On the other hand, guest traffic originating
! the fabric and now blackholed:
!
BR2-Edge1# ping vrf 50 208.67.222.222
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 208.67.222.22
.....
Success rate is 0 percent (0/5)
BR2-Edge1# show sdwan policy service-path vpn 5
source-ip 10.50.102.1 dest-ip 208.67.222.222 pr
Number of possible next hops: 1
Next Hop: Blackhole
```

Use Case 11 Review

In this use case, you are beginning to see the true power of centralized data policies: They enable you to specify different forwarding behaviors on an app-by-app basis. While the forwarding decisions in this policy were made on the basis of a Layer 7 application definition, centralized data policies can match on a number of different Layer 3, Layer 4, and/or Layer 7 criteria. For example, this policy could be extended to match on a combination of the Webex application and a source data prefix, such that the rule would only apply to users residing in certain subnets. Users outside those specific source ranges would follow the traditional forwarding pattern.

This use case also continues to build on the **nat use-vpn 0** action with the **nat fallback** action. With this combination of actions, network administrators can create dynamic policies to meet business needs, while at the same time being able to create predictable behavior in the event of a failure. If network administrators are concerned about conserving limited site-to-site bandwidth, the configurations in this policy can easily be extended to provide access for many different applications using local Internet egress points. These administrators could then be very selective about which applications would be permitted to fall back to the limited site-to-site tunnels, and NAT DIA tracker could be configured on the NAT-enabled interface. For example, a cloud-hosted payroll application might be considered business critical and permitted to fall back to the site-to-site path. At the same time, Internet radio may be permitted to use the local Internet egress path, but in the event of a

Technet24

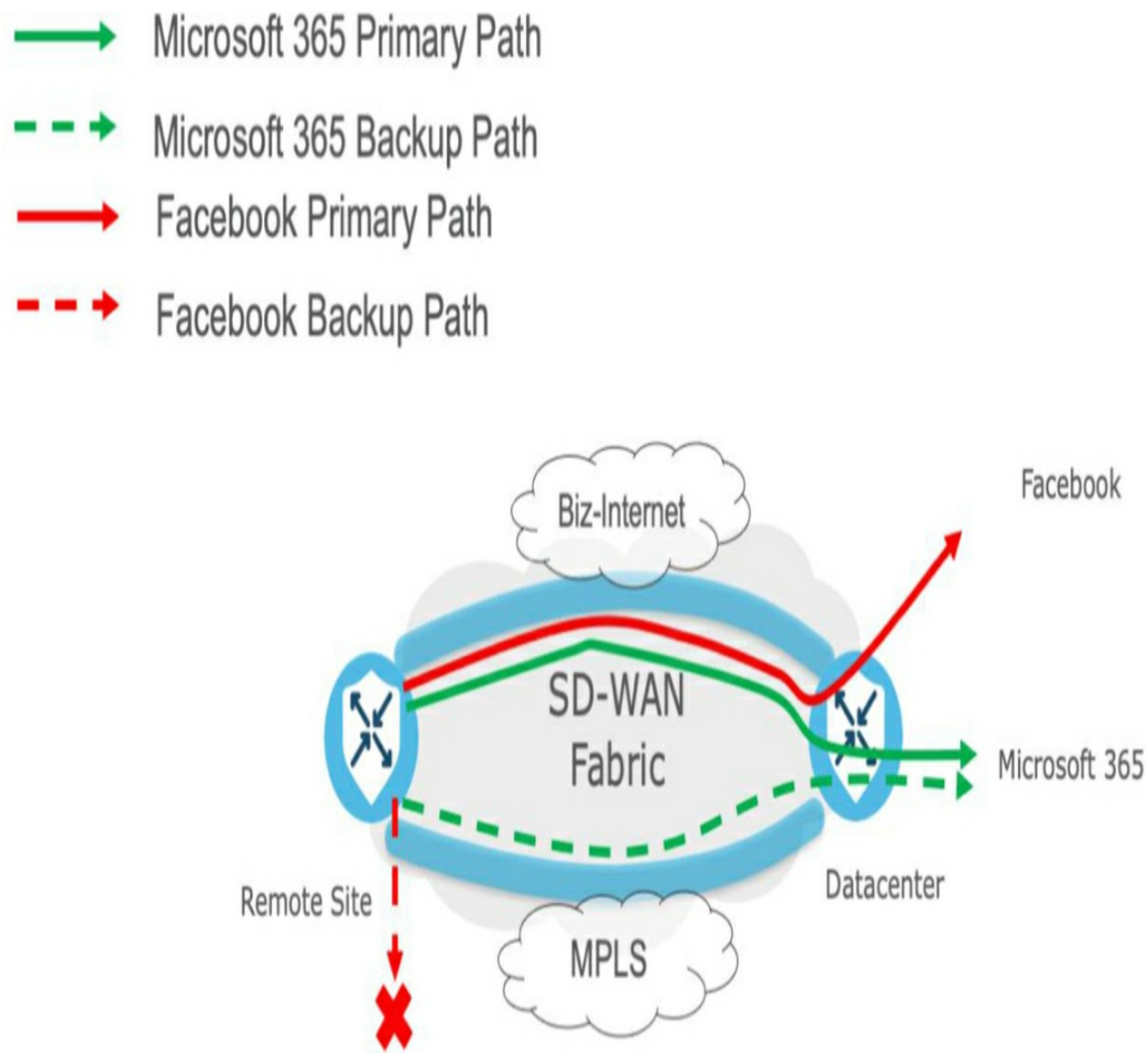
failure, it would not be considered business critical and would not be permitted to fall back to the site-to-site path.

Use Case 12: Application-Based Traffic Engineering

Enterprise WANs are becoming increasingly important to business and are constantly being tasked with providing faster and more reliable connectivity, all while using less expensive transport networks and cutting overall costs. Traditionally, organizations used expensive leased-line transports with small bandwidth but guaranteed service-level agreements (SLAs) in order to connect their diffuse sites. As organizations transition to hybrid transport environments (where they may still have expensive small links that are being augmented with substantially larger circuits without guaranteed SLAs), there is an ever-growing need for new and powerful tools to be able to dictate which paths applications take as they flow across the WAN.

As Figure 8-31 from Use Case 11 shows, the currently configured policy has both critical and noncritical business traffic, such as Microsoft 365 or Facebook traffic, indistinctly load-shared across the MPLS and BIZ-INTERNET paths. Wasting the limited and expensive MPLS bandwidth on noncritical traffic is not desired, so in this use case, the centralized data policy will continue to be refined so that certain classes of traffic are forwarded across the BIZ-INTERNET path only. This policy will consider two different classes of traffic, Microsoft 365 and Facebook traffic, and set different forwarding rules for each. Microsoft 365 traffic will be preferred

across the BIZ-INTERNET SD-WAN tunnels. In the event that the BIZ-INTERNET tunnels are not available, the Microsoft 365 traffic will move to any of the other transports available. Traffic for the second application, Facebook, will be required to use the BIZ-INTERNET tunnels to reach the data centers. In the event that the BIZ-INTERNET tunnels are not available, this application will be unavailable. These traffic patterns are illustrated in Figure 8-33.



Technet24

Figure 8-33 *Desired Forwarding Behavior for Facebook and Microsoft 365 Traffic*

Before the policy from Example 8-4 has been modified, the forwarding behavior shown in Example 8-6 can be observed.

Example 8-6 *Application Forwarding Behavior Without Policy Changes*

```
!
! Before modifications are made to the policy,
traffic is
! forwarded across both links.
!
BR2-Edge1# sh sdwan policy service-path vpn 10
10.10.102.1 dest-ip 208.67.222.222 protocol 1 a
Number of possible next hops: 3
Next Hop: IPsec
  Source: 209.165.201.6 12346 Destination: 209.
biz-internet Remote Color: biz-internet Remote
Next Hop: IPsec
  Source: 209.165.202.130 12346 Destination: 20
Remote Color: biz-internet Remote System IP: 10
Next Hop: IPsec
  Source: 192.168.20.6 12346 Destination: 192.1
```

```
Color: mpls Remote Color: mpls Remote System IP
BR2-Edge1# sh sdwan policy service-path vpn 10
10.10.102.1 dest-ip 208.67.222.222 protocol 1 a
Number of possible next hops: 3
Next Hop: IPsec
  Source: 209.165.201.6 12346 Destination: 209.
biz-internet Remote Color: biz-internet Remote
Next Hop: IPsec
  Source: 209.165.202.130 12346 Destination: 20
Remote Color: biz-internet Remote System IP: 10
Next Hop: IPsec
  Source: 192.168.20.6 12346 Destination: 192.1
Color: mpls Remote Color: mpls Remote System IP
BR2-Edge1#
```

Example 8-6 again uses the **show sdwan policy service-path** command to determine the path or paths that particular flows would take through a WAN Edge router. This example shows the outputs for the application facebook and the application ms-office-365, indicating that flows from both applications would be load-shared across the MPLS, LTE, and BIZ-INTERNET tunnels to DC1-Edge1. In order to change this behavior, the policy configured in Example 8-7 uses the **local-tloc** action and the **tloc-list** action.

Technet24

The **local-tloc** action is configured with this syntax:

```
set local-tloc color {color} [encap {ipsec|gre}]
```

where *color* is any one of the supported TLOC colors. This action directs packets to be forwarded out the local TLOC that is specified in the **color** argument. If this TLOC is not available (because, for example, it is not configured or the tunnel is down), the traffic is forwarded out any valid TLOC, as indicated by the routing table. In addition, the configuration command **set local-tloc-list** allows for the selection of one or more colors. SD-WAN Manager defaults to using this syntax when policies are configured through the GUI.

Similar to what **set local-tloc-list** offers when listing more than one preferred transport, the command **set preferred-color-group** {*color-group-name*} not only helps select the egress local TLOC but provides up to three levels of color preference.

The **tloc-list** action in a centralized data policy is similar in structure and function to the centralized control policy **tloc-list** action that we explored in Use Case 2 (refer to Chapter 7). By using the **tloc-list** action, a network administrator is statically specifying the fabric tunnel endpoints toward which the flow will be forwarded. This functionality is roughly equivalent to the **set next-hop-address** functionality in a traditional policy-based-routing route map. With the **tloc-list** action, if the TLOCs specified in the list are not

available, then the traffic is blackholed, even though there may be a different path available.

To summarize, the **local-tloc** action selects the *preferred* egress TLOC on the *local* WAN Edge router, the **preferred-color-group** action provides tiered preference for local egress TLOCs, and the **tloc-list** action *mandates* the TLOCs on the *receiving* WAN Edge that the traffic will be forwarded to. The modified policy in Example 8-7 shows these two different configurations.

Example 8-7 *Application-Based Traffic Engineering Policy*

```
policy
 ! <<<No changes were made to the control polic
 ! the Guest VPN data policies, and they are or
 !
! <<<No changes were made to the control polici
 ! the Guest VPN data policies, and they are or
 !
data-policy _Corporate_Segment_Branch_AppTraffi
  vpn-list Corporate_Segment
    sequence 1
     match
      app-list TRUSTED_APPS
      source-ip 0.0.0.0/0
```

Technet24

```
      !
      action accept
       nat use-vpn 0
       nat fallback
       count CORP_DCA_-1766920491
      !
     !
     !
     ! Sequence 11 uses the local-tloc-list comm
     ! on the local WAN Edge that are PREFERRED
     !
     sequence 11
      match
       app-list office365_apps
       source-ip 0.0.0.0/0
      !
      action accept
       count CORP_M365_-1766920491
       set
       local-tloc-list
        color biz-internet
        encap ipsec
       !
      !
     !
```

```
    !
    ! Sequence 21 uses the tloc-list command to
    ! MUST be forwarded to. If the TLOCs are un
    ! dropped.
    !
    sequence 21
     match
      app-list Facebook
      source-ip 0.0.0.0/0
     !
     action accept
      count CORP_FACEBOOK_-1766920491
     set
      vpn 10
      tloc-list DC_INET_TLOCS
      !
     !
    !
 default-action accept
!
 vpn-list Guest_Segment
<<<Output omitted for brevity>>>
!
lists
! <<<Some lists without changes are omitted fo
```

Technet24

```
 !
  app-list Facebook
   app facebook
   app facebook-audio
   app facebook_apps
   app facebook-video
   app facebook_messenger
   app facebook_mail
   app facebook_live
   app facebook_video
   app fbcdn
   app facebook-media

  !
 app-list office365_apps
  <<<Output omitted for brevity>>>
   app sharepoint
   app ms-office-365
   app outlook-web-service
   app skydrive
<<<Output omitted for brevity>>>
  !
  site-list BranchOffices
   site-id 100-199
  !
```

```
  tloc-list DC_INET_TLOCS
   tloc 10.0.10.1 color biz-internet encap ipse
   tloc 10.0.10.2 color biz-internet encap ipse
  !
 !
!
apply-policy
 site-list Europe_Branches
  control-policy Euro_Reg_Mesh_with_FW_MultiTop
 !
 site-list BranchOffices
  data-policy _Corporate_Segment_Branch_AppBase
  control-policy Branch_Extranet_Route_Leaking
  vpn-membership vpnMembership_373293275
 !
 site-list DCs
  control-policy DC_Inbound_Control_Policy in
 !
 site-list North_America_Branches
  control-policy North_America_Reg_Mesh_with_FW
 !
!
```

Technet24

Note

Keep in mind that Cisco Catalyst SD-WAN often provides multiple ways to achieve the same result. For example, if administrators want to send traffic such as Microsoft 365 traffic directly to the Internet, but need a failover path in case direct Internet access fails, previously covered options like using the NAT VPN in conjunction with **nat fallback** can fulfill the requirement. On the other hand, if noncritical traffic, like Facebook traffic in the current use case, does not need an alternative path due to its lower business priority, the **restrict** action makes it possible to drop undesired traffic.

In sequence 11 in Example 8-7, Microsoft 365 application traffic is being matched, and then the **local-tloc-list** action is configured to preferably forward this traffic out the BIZ-INTERNET TLOC, if the TLOC is available. In sequence 21, Facebook traffic is being matched and is being forwarded to BIZ-INTERNET TLOCs on DC1-Edge1 and DC1-Edge2, as specified with the **tloc-list** action. The effects of these two different configuration options can be seen in Example 8-8, where the results are shown in both a steady state and a state where the Internet connection of both WAN Edge devices at DC1 has failed.



Example 8-8 *Application Traffic Engineering Behavior with Policy in Steady State and Failed State*

```
! When all of the Biz-Internet BFD Sessions to
!
BR2-Edge1#sh sdwan bfd sessions table | include
                                          SRC    DS
TX
SRC IP           DST IP         PROTO  PORT   P
COLOR         STATE  MULTIPLIER  INTERVAL  UPTI
-----------------------------------------------
-----------------------------------------------
-
209.165.201.6    209.165.201.1    ipsec  12346
10    biz-internet  biz-internet  up     7
1:20:16:12  3
209.165.202.130  209.165.201.1    ipsec  12346
10    lte          biz-internet  up     7
1:20:16:04  3
209.165.201.6    209.165.201.2    ipsec  12346
10    biz-internet  biz-internet  up     7
0:00:04:07  5
209.165.202.130  209.165.201.2    ipsec  12346
lte            biz-internet      up     7
```

Technet24

```
!
! In steady state, when all of the links are op
! have the same effect:  Traffic matching the M
matching
! the Facebook app-list are both forwarded acro
! 209.165.201.1    (DC1-Edge1).
BR2-Edge1# show sdwan policy service-path vpn 1
source-ip 10.10.102.1 dest-ip 208.67.222.222 pr
Next Hop: IPsec
  Source: 209.165.201.6 12346 Destination: 209.
biz-internet Remote Color: biz-internet Remote
BR2-Edge1# show sdwan policy service-path vpn 1
source-ip 10.10.102.1 dest-ip 208.67.222.222 pr
 Number of possible next hops: 1
Next Hop: IPsec
  Source: 209.165.201.6 12346 Destination: 209.
biz-internet Remote Color: biz-internet Remote
!
! After an internet failure at DC1, all of the
! Site 10 are down:
!
BR2-Edge1#sh sdwan bfd sessions table | include
                                      SRC   DST
DETECT     TX
SRC IP          DST IP          PROTO  PORT   P
```

```
COLOR         STATE  MULTIPLIER  INTERVAL  UPTI
-------------------------------------------------
-------------------------------------------------
-
209.165.201.6    209.165.201.1    ipsec  12346
10    biz-internet  biz-internet  down     7
1:20:16:12  5
209.165.202.130  209.165.201.1    ipsec  12346
10    lte           biz-internet  down     7
1:20:16:04  5
209.165.201.6    209.165.201.2    ipsec  12346
10    biz-internet  biz-internet  down     7
0:00:04:07  7
209.165.202.130  209.165.201.2    ipsec  12346
lte             biz-internet     down      7
!
! In a failed state, where there is no longer a
color,
! the Microsoft 365 traffic will be forwarded a
!
BR2-Edge1#show sdwan policy service-path vpn 10
GigabitEthernet 4 source-ip 10.10.102.1 dest-ip
protocol 1 app ms-office-365 all
Number of possible next hops: 1
Next Hop: IPsec
```

Technet24

```
  Source: 192.168.20.6 12346 Destination: 192.1
Remote Color: mpls Remote System IP: 10.0.10.1
! The Facebook application, will not failover t
! action statically specifies the next-hop tunn
! endpoints are not available, the traffic is b
!
BR2-Edge1#show sdwan policy service-path vpn 10
GigabitEthernet 4 source-ip 10.10.102.1 dest-ip
protocol 1 app facebook all
Number of possible next hops: 1
Next Hop: Blackhole
BR2-Edge1#
```

When the network is in a state where all of BIZ-INTERNET tunnels are operating as expected, Example 8-8 shows that the forwarding behavior for these two configurations is the same: Both the Microsoft 365 and Facebook flows are forwarded across the BIZ-INTERNET tunnel to 209.165.201.1, the Internet interface IP address of DC1-Edge1. The difference between these two configurations becomes apparent when the network is in a failed state, as would occur if the BIZ-INTERNET transport to DC1 were severed by a fiber-seeking backhoe. Example 8-8 shows that the Microsoft 365 application, configured with the **local-tloc-list** action, fails over to the MPLS TLOC and continues to operate. The Facebook application, on the other hand, which was configured with the **tloc-list** action, where the next hops

were statically specified, does not fail over, and the traffic is instead blackholed.

Application-Based Traffic Engineering with Policy Groups

Let's look at an example of the same traffic engineering scenario we've been working with but now leveraging policy groups. Create a policy group by going to Configuration > Policy Groups > +Add Policy Group and providing a name and description, as shown in Figure 8-34.

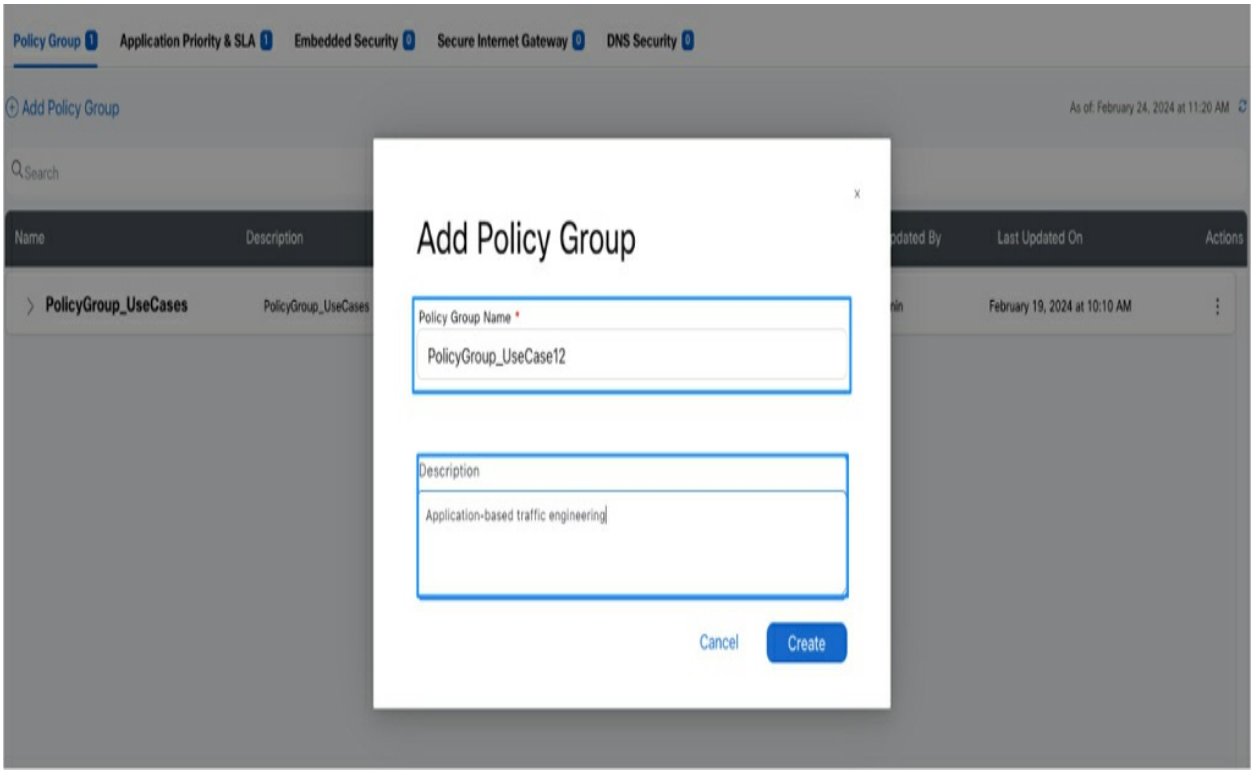


Figure 8-34 *Creating a Policy Group*

Next, create an Application Priority profile, by clicking Create New and providing a name and description, as shown in Figure 8-35.

Technet24

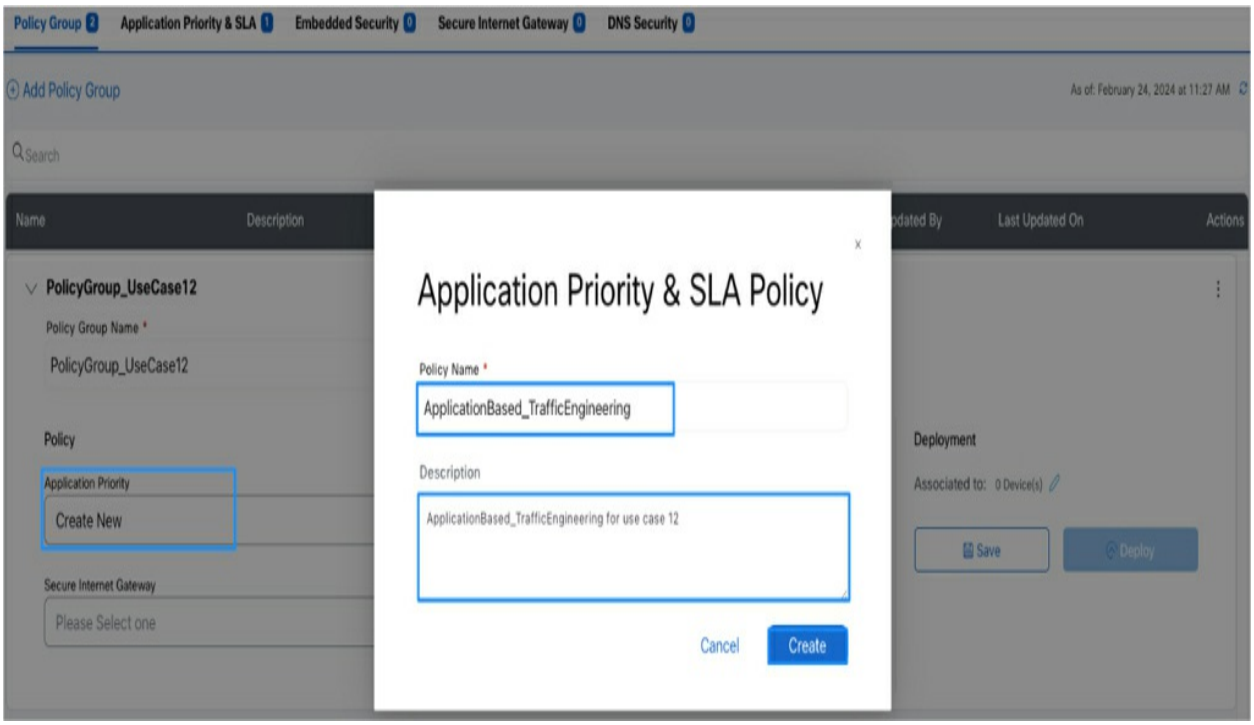


Figure 8-35 *Creating an Application Priority & SLA Policy*

In the Application Priority profile you just created, click Advanced Layout, then select Add Traffic Policy and specify the name, the VPN (Corporate_Segment), and the direction of the policy (Service), as shown in Figure 8-36.

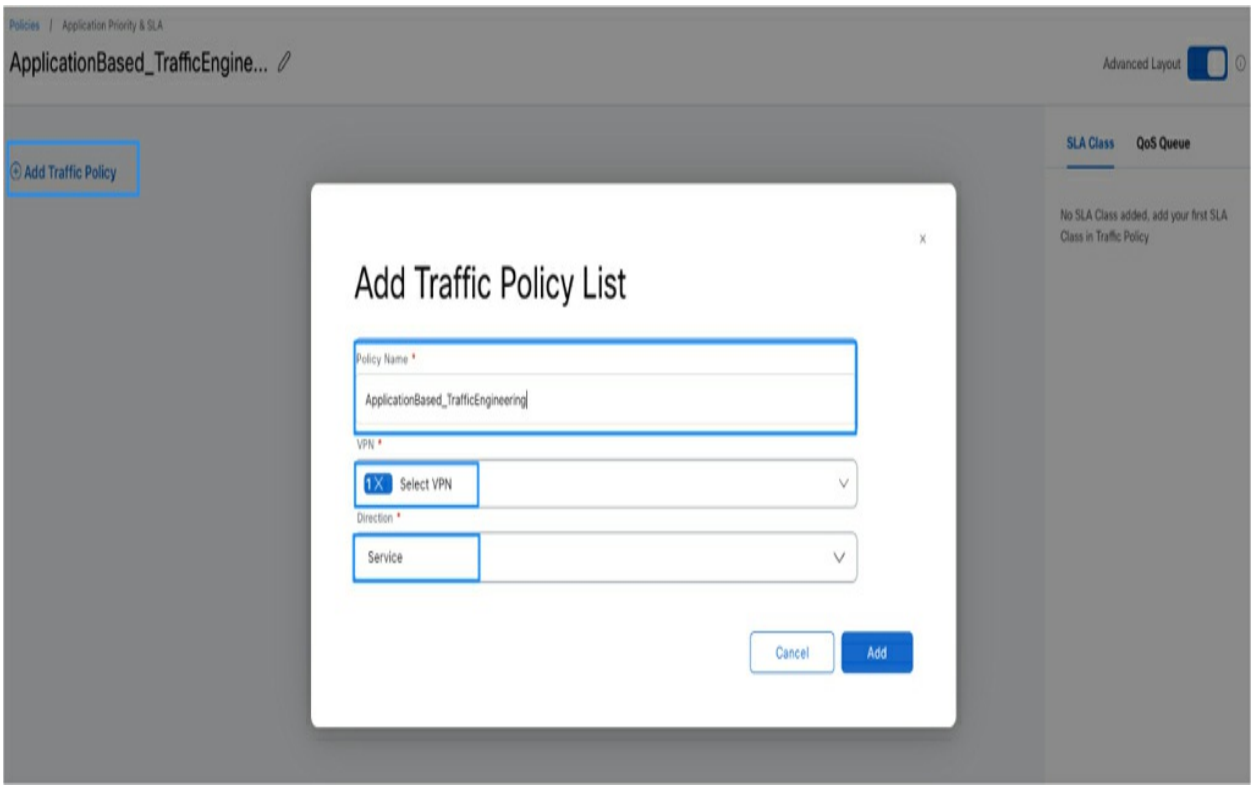


Figure 8-36 *Creating a Traffic Policy for VPN 10*

Just as with the conventional policy methodology, two sequence rules are created, one for each of the applications discussed. Click +Add Rules and use this first rule for Microsoft 365 traffic that is matched based on the Application / Traffic Class criteria, and then click Create New and select the desired applications under Application-Family to create the list. Next, select Accept under Base Action. Under LOCAL TLOC, select biz-internet as the color and IPsec as the encapsulation to be used for the preferred path (see Figure 8-37). Then click Save Match and Actions.

Technet24

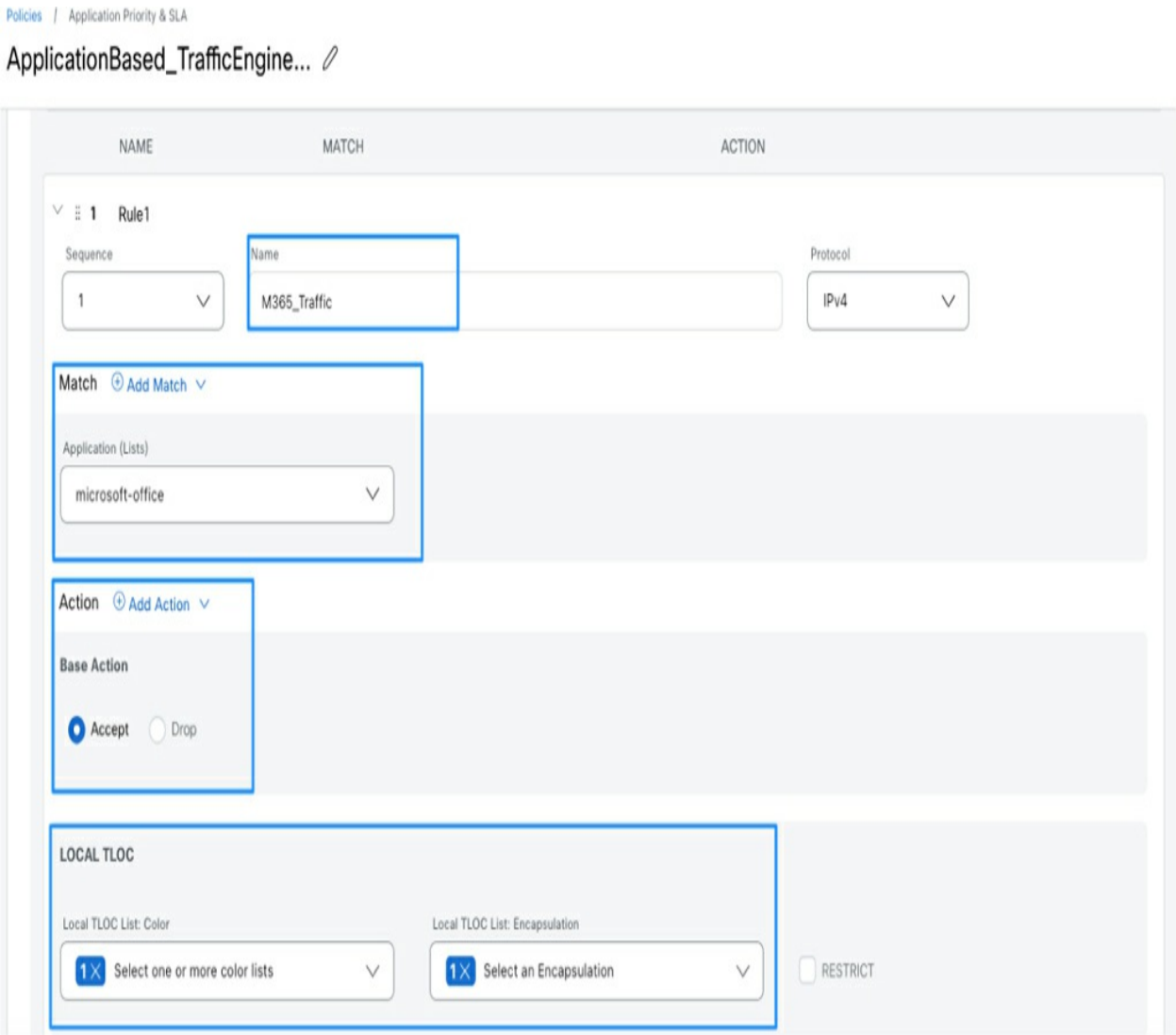


Figure 8-37 *Defining a Rule for Microsoft 365 Traffic*

Next, click +Add Rules to add a second sequence for Facebook traffic, which is matched based on the application list with facebook as the value. Then select Accept under Base Action. Under TLOC List, select the BIZ-INTERNET TLOCs at DC1-Edge1 and DC1-Edge2 as the preferred paths (see Figure 8-38). (This rule differs from the previous rule in that unavailability of these paths will drop the traffic.) Click Save Match and Actions again.

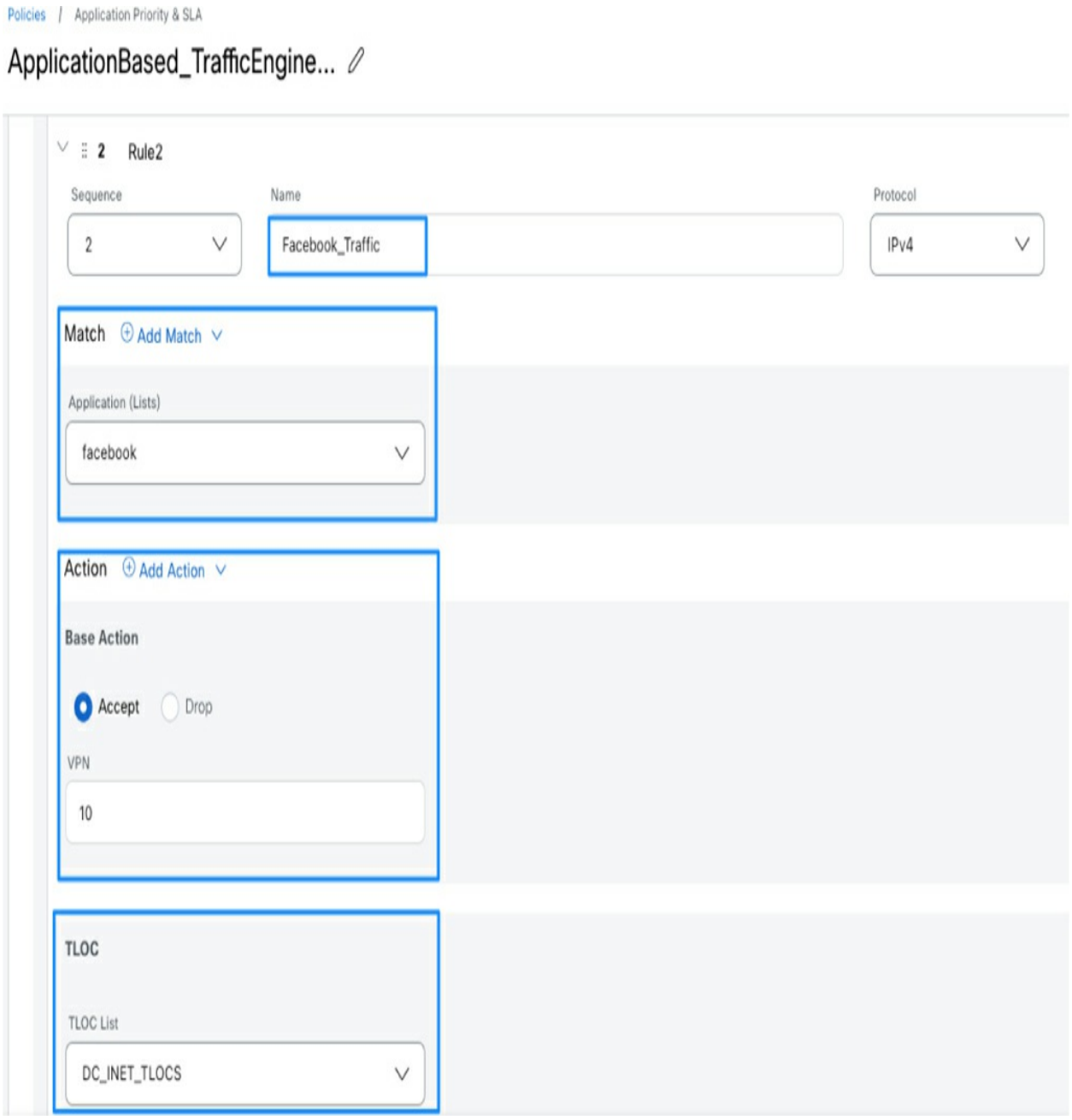


Figure 8-38 *Defining a Rule for Facebook Traffic*

After the rules specifying the actions for the different types of traffic are set, click Save, as shown in Figure 8-39.

Technet24

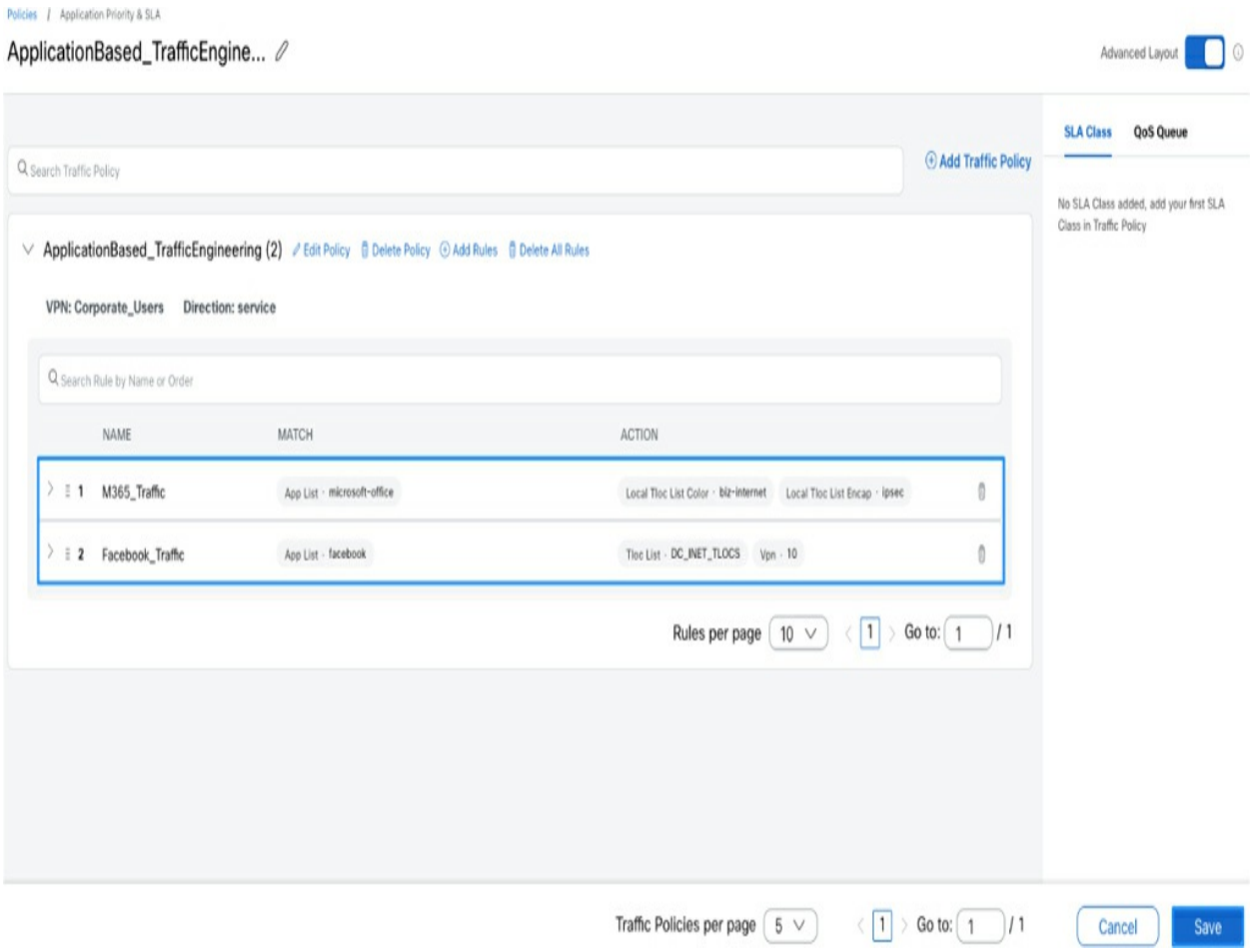


Figure 8-39 *Application Priority Policy Profile Preview*

Finally, deploy the policy group to the desired devices. The workflow guides you through the required steps to define the devices that will be subject to one or more policy profiles contained in the policy group. In this case, only the newly created Application Priority & SLA policy is sent to the devices (see Figure 8-40).

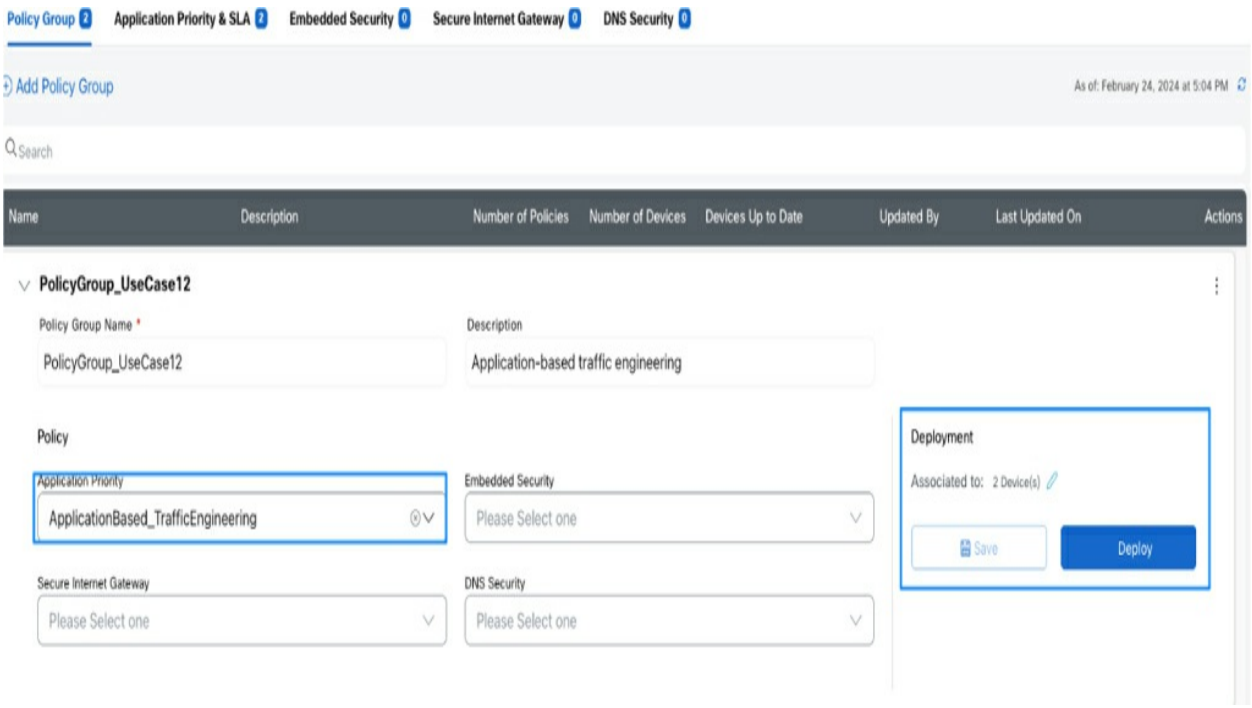


Figure 8-40 *Deploying a Policy Group*

Use Case 12 Review

This use case again illustrates that there are often several different design and configuration choices that can be used to accomplish any specific task. It is important for administrators to not only consider the forwarding behavior that is desired but also to think through the possible outcomes and effects of different failure scenarios and the effects of design choices on the eventual outcomes. As Example 8-8 shows, TLOC lists are powerful but unforgiving tools; use them with caution.

This example uses Facebook and Microsoft 365 as two sample applications for illustration purposes, but these placeholders can easily be replaced with applications that are important to the business: thin clients, enterprise

Technet24

resource planning (ERP) systems, file servers, email, collaboration software, and so on. Network administrators can apply the lessons learned here to solve the challenges facing their organizations.

Use Case 13: Protecting Corporate Users with a Secure Internet Gateway

Throughout this chapter, we have explored use cases where users were accessing the Internet by egressing directly from the WAN Edge, such as the Webex application in Use Case 11. In Use Case 12, you saw examples of traffic engineering policies that allowed users to access Internet-based applications by backhauling the traffic through a data center. This use case explores a third option: integrating a cloud-delivered firewall (CDFW), such as Cisco Umbrella Secure Internet Gateway (SIG), to the SD-WAN fabric and redirecting traffic flows to Cisco Umbrella SIG. This use case focuses on the integration of Umbrella SIG with the SD-WAN fabric and the centralized data policies that are necessary to redirect traffic to Umbrella SIG.

Note

For more information about Cisco Umbrella SIG and the specific configurations for integrating Umbrella with Cisco SD-WAN, see the references listed at the end of this chapter and visit https://umbrella.cisco.com.

In Chapter 7, you saw a firewall inside one of the sites configured and advertised as a service in the SD-WAN fabric. All the traffic that was passing between sites was forwarded through the service based on manipulation of the routing table with a centralized control policy. In this use case, rather than manipulating the routing table to send all traffic through the CDFW, a centralized data policy will be created to send only a specific set of applications through the firewall—specifically, a variety of Google apps. Figure 8-41 illustrates this traffic flow.

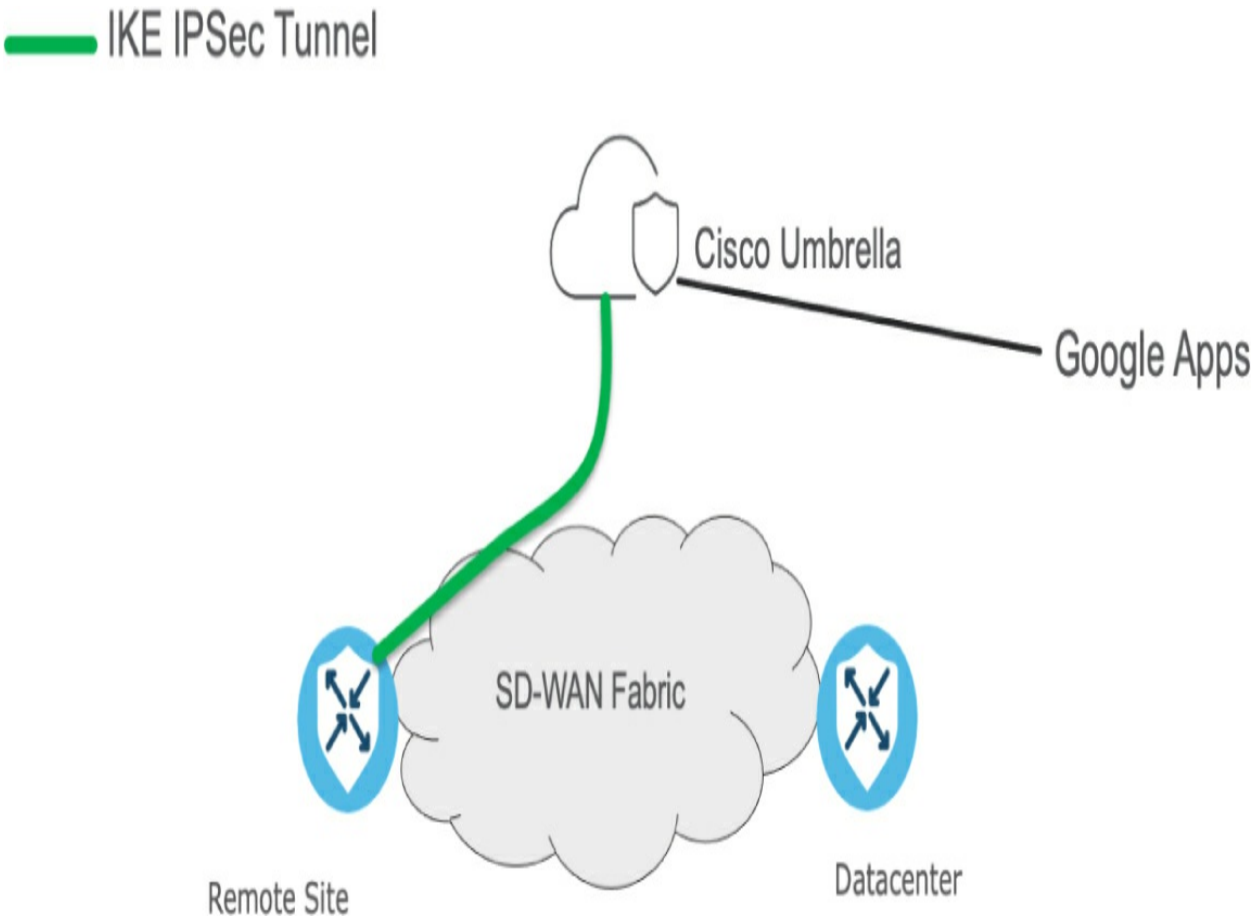


Figure 8-41 *Redirecting Google Apps Traffic to Cisco Umbrella SIG*

The first step in the process is to establish the tunnel for connectivity to the

Technet24

Secure Internet Gateway solution, as shown in Example 8-9. While this example uses Cisco Umbrella, this integration would also work with another SIG provider that uses IPsec or GRE tunnels to establish connectivity.

Example 8-9 *SD-WAN Router Configuration for a Secure Internet Gateway Tunnel*

```
!
! An IPSec tunnel to Cisco Umbrella is configu
!
BR2-Edge1#sh sdwan running-config  | section Tu
interface Tunnel100001
 no shutdown
 ip unnumbered      GigabitEthernet2
 no ip clear-dont-fragment
 ip mtu             1400
 tunnel source GigabitEthernet2
 tunnel destination dynamic
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile if-ipsec1-ipse
 tunnel vrf multiplexing
 tunnel route-via GigabitEthernet2 mandatory
   interface-pair Tunnel100001 active-interface
backup-interface-weight 1
```

```
 interface Tunnel100001
  tunnel-options tunnel-set secure-internet-gat
tunnel-dc-preference primary-dc source-interfac
BR2-Edge1#
BR2-Edge1#sh sdwan running-config | section sig
 service sig vrf global
  ha-pairs
   interface-pair Tunnel100001 active-interface
backup-interface-weight 1
  !
BR2-Edge1#show sdwan secure-internet-gateway  t
TUNNEL IF
TRACKER  SITE  DESTINATION           TUNNEL
NAME         TUNNEL ID  TUNNEL NAME
STATE    ID   DATA CENTER   PROVIDER  TYPE    T
-----------------------------------------------
-----------------------------------------------
---------
Tunnel100001  622182010  SITE102SYS10x0x102x1IK
Enabled  102   CoreSite NY2  Umbrella  IPsec
BR2-Edge1#
```

In Example 8-9, the status of the SIG tunnels is also validated. The same verification is available at Monitor > Tunnels > SIG Tunnels in SD-WAN

Technet24

Manager (see Figure 8-42). The details for establishing these tunnels are typically provided by the SIG provider. After the SIG tunnel is configured, a centralized data policy needs to be created to redirect the traffic of interest to the security service.

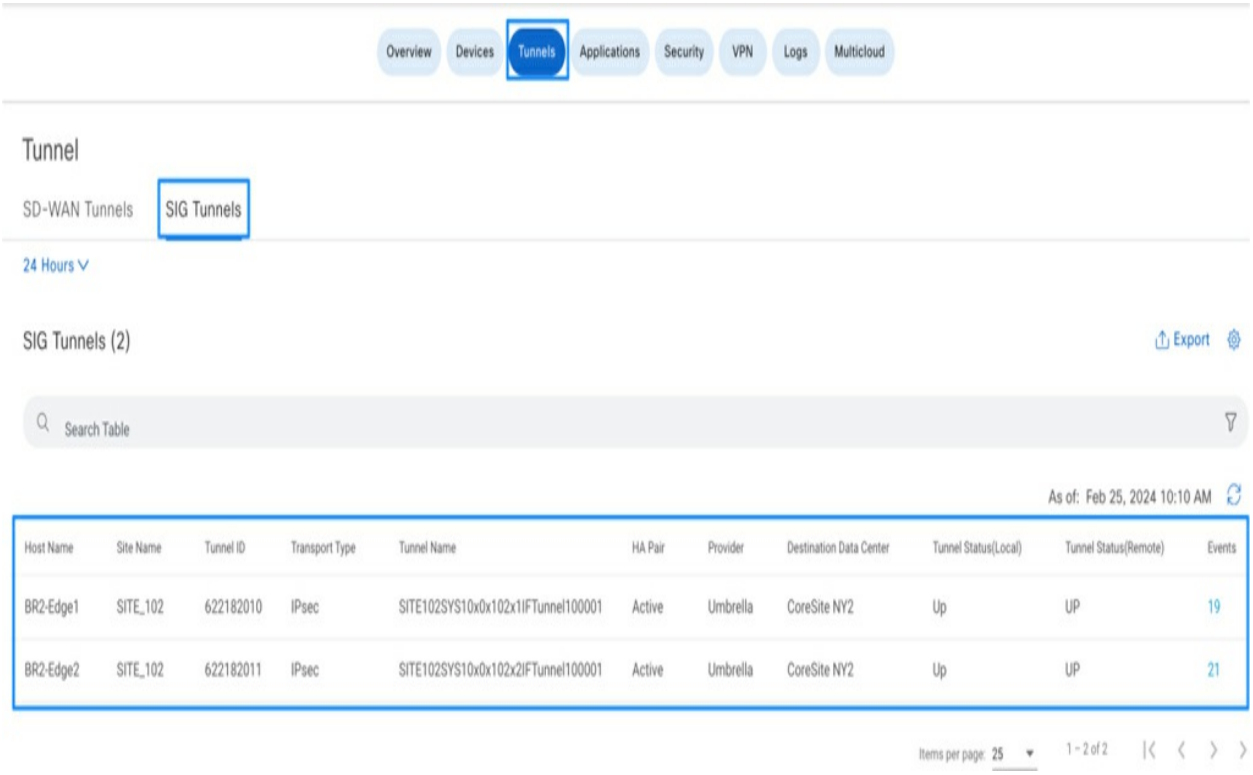


Figure 8-42 *Monitoring the SIG Tunnel from SD-WAN Manager*



Centralized control policies are enforced centrally on the SD-WAN Controller and are only evaluated when sending or receiving control plane updates with OMP. Therefore, there is limited impact from the control policies on the SD-WAN Controllers, and there is no performance impact on the WAN Edge routers. Centralized data policies, however, are forwarded to

the WAN Edge routers from the SD-WAN Controller through OMP, where each flow is then evaluated against the policy. Hence, the structure of the policy can have significant performance implications. It is beneficial to structure a policy such that as few of the sequences as necessary are evaluated for each flow. For example, if you have a data policy that is only concerned with external applications, you can create the first sequence of the data policy to match all the internal traffic and eliminate it from consideration for the rest of the sequences in the policy (rather than relying on the default action at the end of the data policy).

While not strictly necessary for this policy in a lab environment, this best practice has been implemented in the policy in Example 8-10 by the first sequence in the policy.

Example 8-10 *SIG Policy*

```
policy
 ! <<<No changes were made to the control polic
 ! the Guest VPN data policies, and they are or
 !
 data-policy
_Corporate_Segment_Corporate_DCA_SIG_Guest_Segr
    ! Sequence 1 stops any traffic that is beir
    ! needing to be processed by any of the rul
```

Technet24

```
    ! applications.
    sequence 1
     match
      destination-data-prefix-list INTERNAL_ADD
     !
     action accept
      count INTERNAL_PCKTS_1774762089
     !
    !
    sequence 11
     match
      app-list TRUSTED_APPS
      source-ip 0.0.0.0/0
     !
     action accept
      nat use-vpn 0
      nat fallback
      count CORP_DCA_1774762089
     !
    !
    sequence 21
     match
      app-list office365_apps
      source-ip 0.0.0.0/0
     !
```

```
      action accept
       count CORP_M365_1774762089
       set
        local-tloc-list
         color biz-internet
         encap ipsec
       !
      !
     !
     sequence 31
      match
       app-list Facebook
       source-ip 0.0.0.0/0
      !
      action accept
       count CORP_FACEBOOK_1774762089
       set
        vpn 10
        tloc-list DC_INET_TLOCS
       !
      !
     !
     !
     ! Sequence 41 redirects applications matchi
SIG tunnel
```

Technet24

```
    !
    sequence 41
     match
      app-list Google_Apps
      source-ip 0.0.0.0/0
     !
     action accept
      count UMBRELLA_PCKTS_1041684049
      sig
      !
     !
    !
  default-action accept
 !
 vpn-list Guest_Segment
  ! <<<Output omitted for brevity>>>
 !
 lists
 ! <<<Some lists without changes are omitted fo
 !
 !
 ! At SD-WAN Manager some smart default lists a
 ! office365_apps and Google_Apps for example.
of different applications that are
 ! produced by the two organizations. Custom ap
```

```
 ! only a subset of apps, but the default app-l
 ! entire list is not displayed for brevity.
 !
  app-list Google_Apps
   app gmail
   app google
   app google_translate
   app gmail_drive
   app gtalk
   app youtube
   app youtube_hd
! <<< Omitted for brevity >>>
  !
  app-list TRUSTED_APPS
   app webex-meeting
   app webex_weboffice
   app webex
  !
  app-list office365_apps
! <<< Omitted for brevity>>>
   app word_online
   app excel_online
   app onedrive
   app yammer
   app sharepoint
```

Technet24

```
    app ms-office-365
   !
   data-prefix-list INTERNAL_ADDRESSES
    ip-prefix 10.0.0.0/8
    ip-prefix 172.16.0.0/12
    ip-prefix 192.168.0.0/16

   !
  !
 !
 apply-policy
  site-list Europe_Branches
   control-policy Euro_Reg_Mesh_with_FW_MultiTop
  !
  site-list BranchOffices
   data-policy
 _Corporate_Segment_Corporate_DCA_SIG_Guest_Segr
  from-service
   control-policy Branch_Extranet_Route_Leaking
   vpn-membership vpnMembership_1718052183
  !
  site-list DCs
   control-policy DC_Inbound_Control_Policy in
  !
  site-list North_America_Branches
```

```
    control-policy North_America_Reg_Mesh_with_FW
   !
  !
```



Example 8-10 shows the changes that were made to the policy in order to accomplish the objectives of this use case. In sequence 41, packets that match the Google_Apps app list are forwarded to the CDFW with the action **sig**, which enables the service to redirect the specific traffic through the previously established IPsec tunnel. Much as with **nat fallback** in Use Case 11, when SIG integration is used, the Fallback to Routing setting is available, and it specifies that if the local paths to the SIG provider are unavailable, the traffic should not be dropped but should instead follow the routing table forwarding decisions. For additional details and prerequisites, refer to "Cisco Catalyst SD-WAN Security Configuration Guide."

With the policy changes in Example 8-10 applied to the network, you see the results shown in Example 8-11.

Example 8-11 *Validating Redirection to the Secure Internet Gateway Service*

```
BR2-Edge1# show sdwan policy data-policy-filter
```

Technet24

```
 data-policy-filter
_Corporate_Segment_Corporate_DCA_SIG_Guest_Segr
 data-policy-vpnlist Guest_Segment
  data-policy-counter default_action_count
   packets 0
   bytes   0
  data-policy-counter GUEST_DIA_PKTS_-194205008
   packets 0
   bytes   0
  data-policy-counter GUEST_DROPPED_PKTS_-19420
   packets 0
   bytes   0
 data-policy-vpnlist Corporate_Segment
  data-policy-counter CORP_DCA_1774762089
   packets 0
   bytes   0
  data-policy-counter CORP_M365_1774762089
   packets 129941
   bytes   33472283
  data-policy-counter default_action_count
   packets 23303
   bytes   1587839
  data-policy-counter CORP_FACEBOOK_1774762089
   packets 631
   bytes   64413
```

```
   data-policy-counter INTERNAL_PCKTS_1774762089
    packets 50
    bytes   5494!
! The counters indicate that traffic is being
!
data-policy-counter UMBRELLA_SIG_PCKTS_1774762
   packets 3086
   bytes   360261
BR2-Edge1#
```

Example 8-11 shows the use of **show sdwan policy data-policy-filter** to validate that the policy is successfully redirecting traffic to the Cisco Umbrella SIG service.



Figure 8-43 and Figure 8-44 show another good way to get deep insights into the network. These figures show Network Wide Path Insights (NWPI) being leveraged to visualize how traffic from Google apps is forwarded through a SIG tunnel. NWPI provides end-to-end path tracing from SD-WAN Manager, with detailed information from the network level to packet level. For more details, see Chapter 14, "Cisco Catalyst SD-WAN Monitoring and Operations."

Technet24

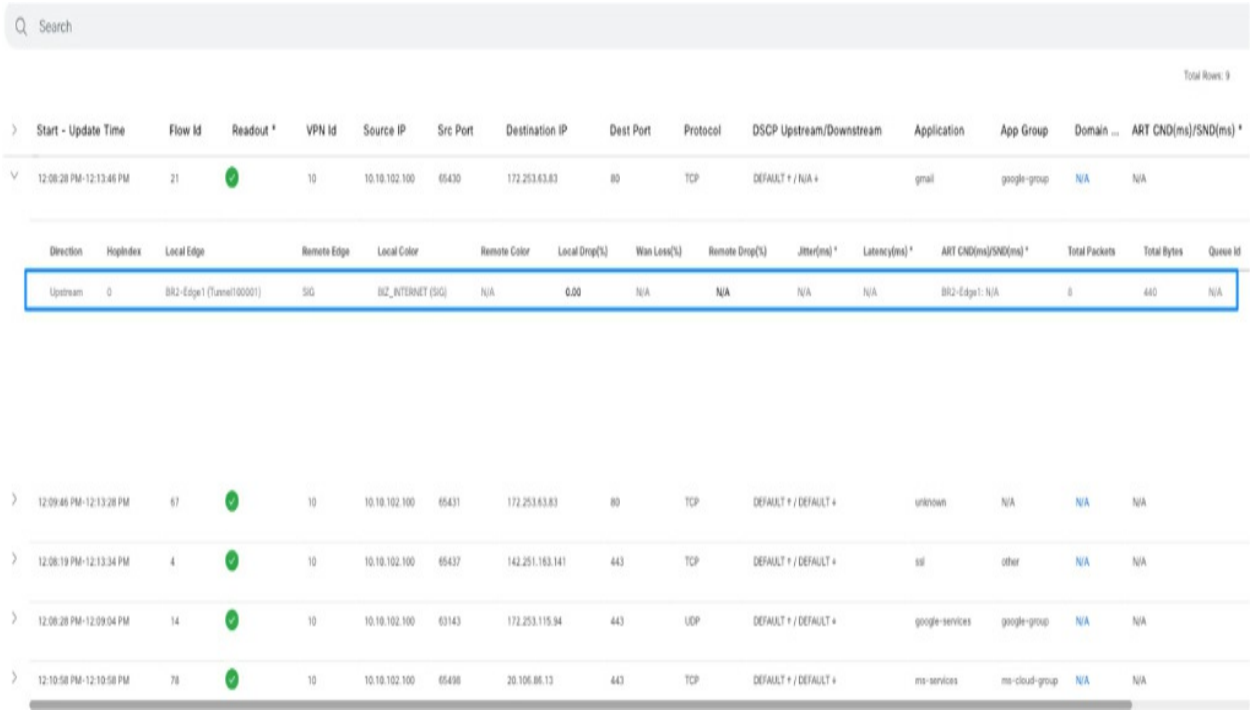


Figure 8-43 *Insight for Gmail from SD-WAN Manager*

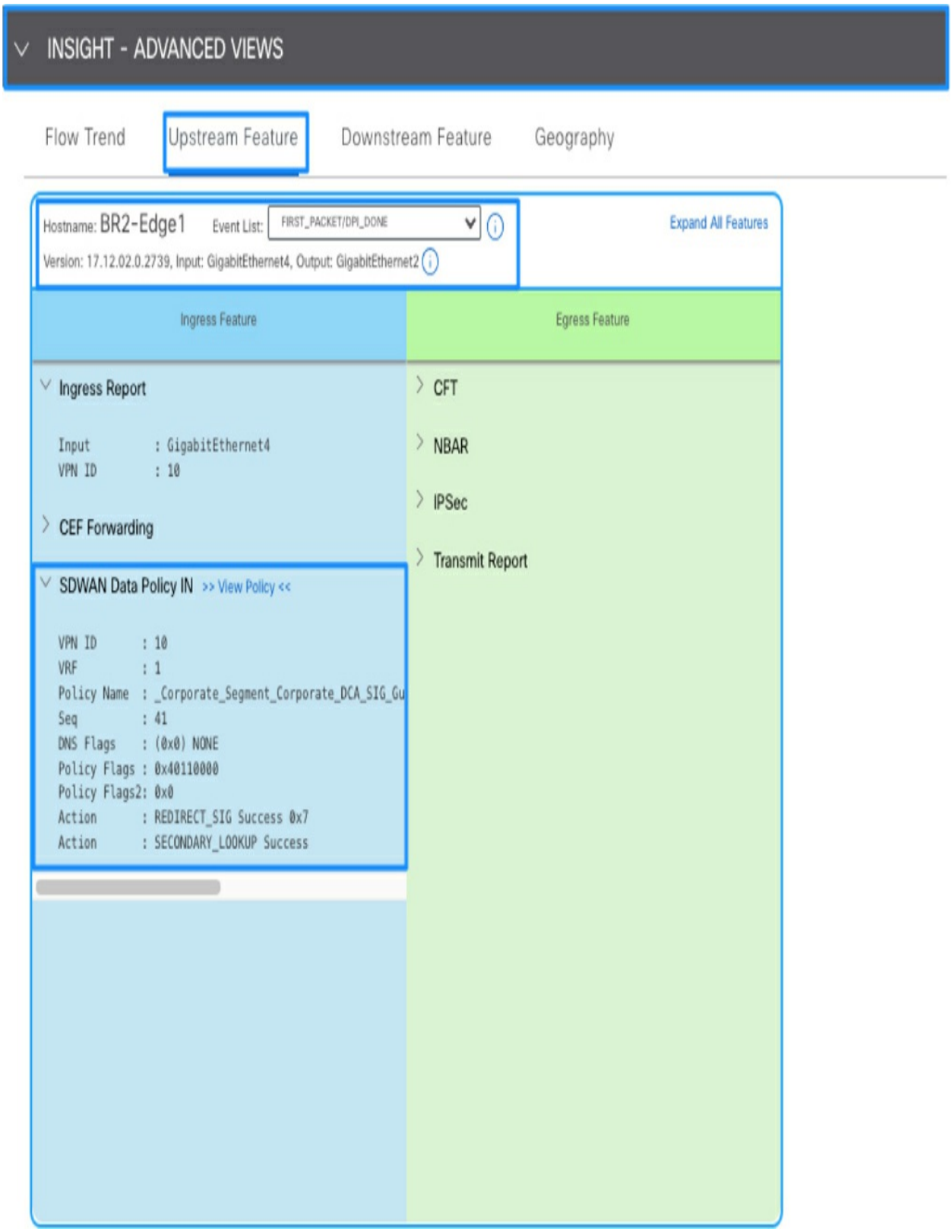


Figure 8-44 *Advanced View Showing a SIG Action for Google Apps*

Technet24

Protecting Corporate Users with a Secure Internet Gateway Using Policy Groups

As previously discussed, Policy Groups offer a simpler way to implement different traffic policies. In this case, we look at using them to create a tunnel to Cisco Umbrella SIG. This first step in this configuration is adding a name and description for the Secure Internet Gateway policy profile, as shown in Figure 8-45.

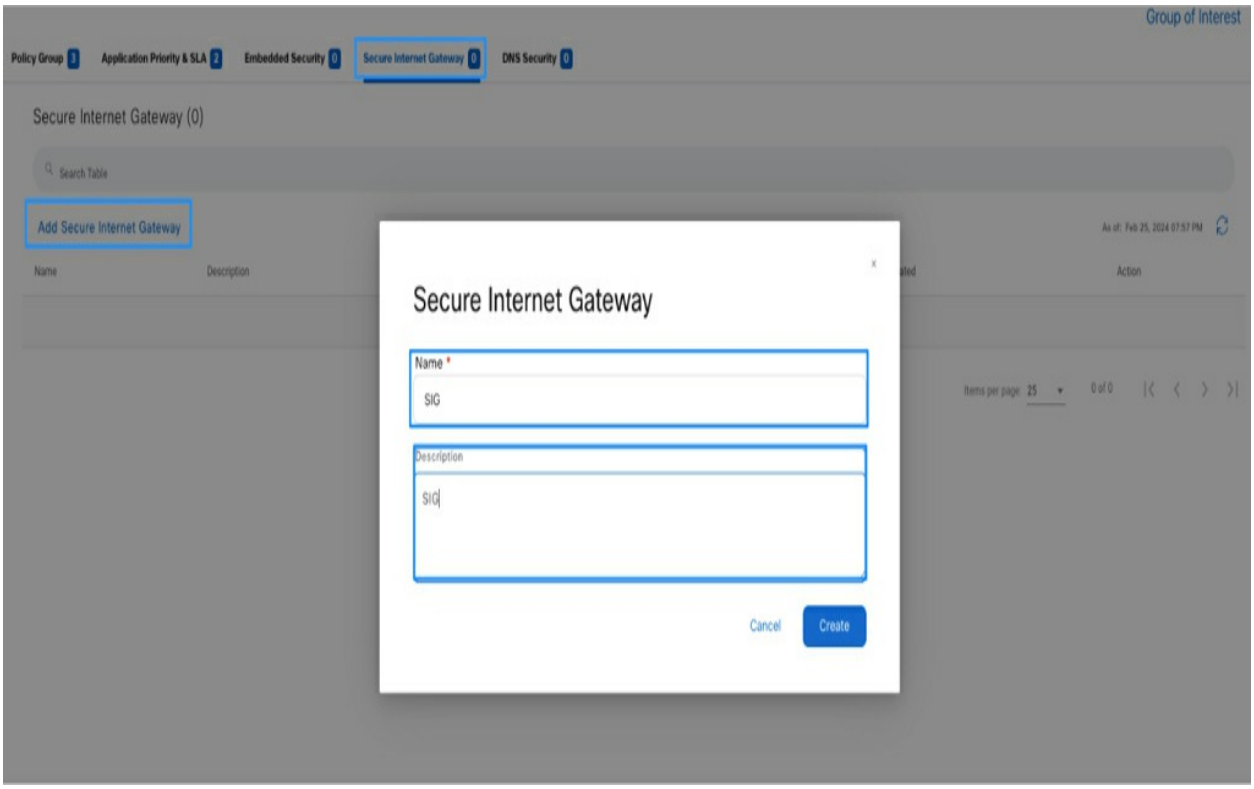


Figure 8-45 *Creating a Secure Internet Gateway Policy Profile*

The next step is to select the SIG provider and provide either the credentials if this is a fresh deployment or the tunnel number and the physical or logical interface where the IPsec tunnel will source from if credentials have been

provided before (see Figure 8-46).

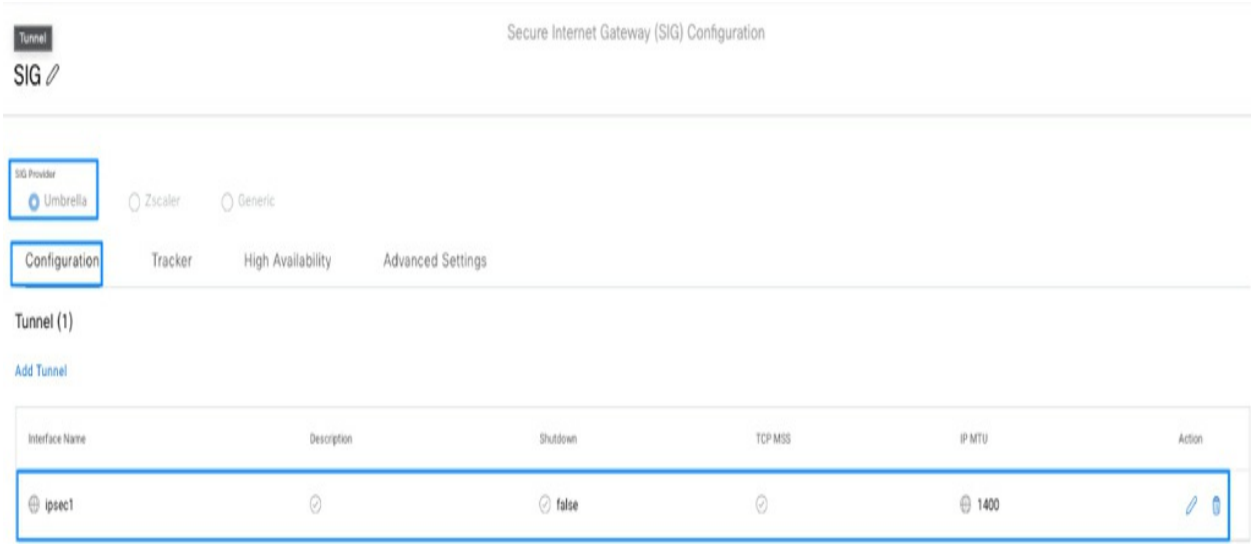


Figure 8-46 *SIG Tunnel Configuration Screen*

A tracker is added by default if the workflow automates the tunnel's creation. An unused private address is requested for this step, but customized trackers can be leveraged as well, as shown in Figure 8-47.

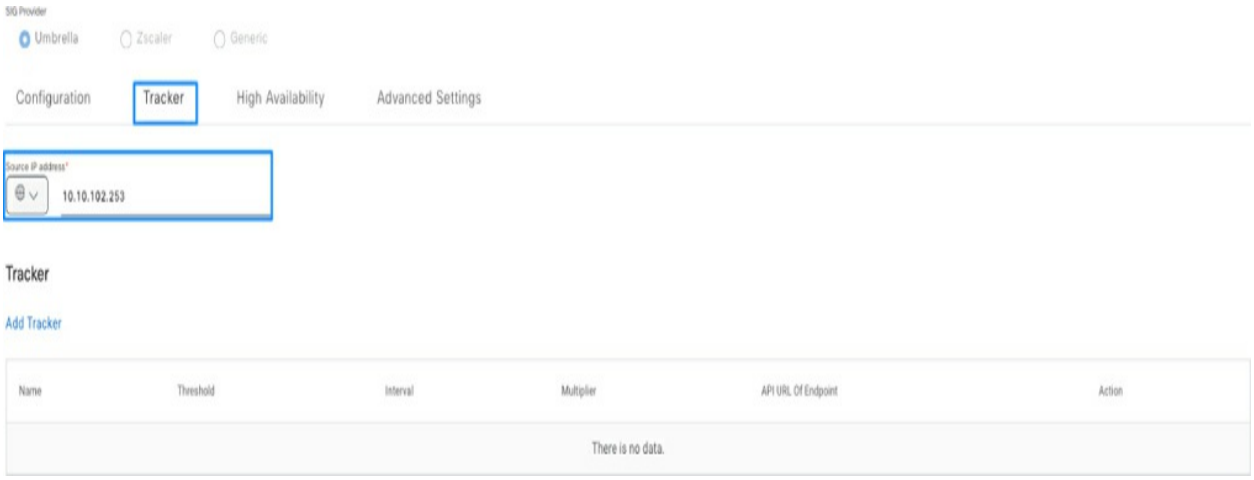


Figure 8-47 *SIG Tunnel Tracker Configuration*

Figure 8-48 shows how active and backup tunnels as well as the weight for

Technet24

each of them can be set. In this case, a single tunnel is configured.

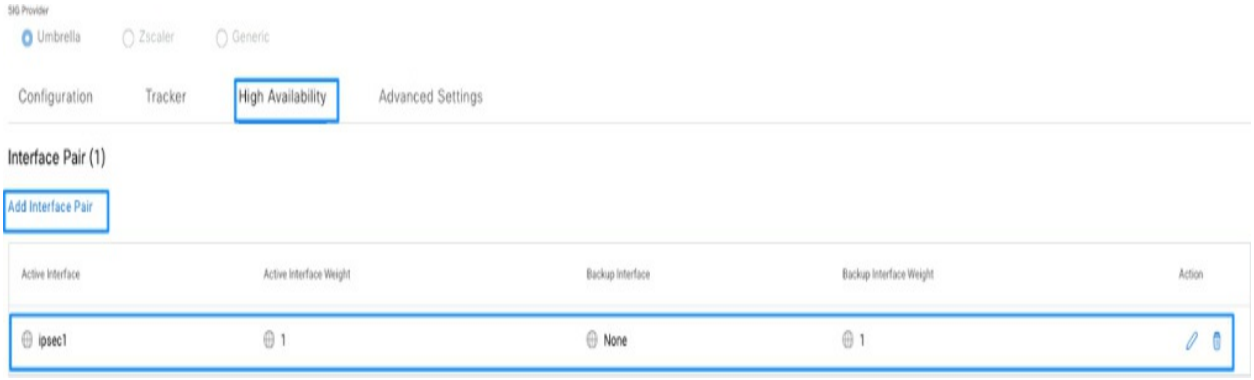


Figure 8-48 *SIG Tunnel Interface Pair Configuration*

Figure 8-49 shows how to allow the user to configure the primary and secondary data centers to send the traffic to. This step is only necessary when location compliance rules apply to an organization or simply when an administrator is seeking predictable behavior.



Figure 8-49 *Advanced Settings for a SIG Tunnel*

As discussed earlier in this chapter, the selection of the policy type, such as SIG, and subsequent deployment to the associated devices is the final step (see Figure 8-50). This configuration deals with tunnel establishment.

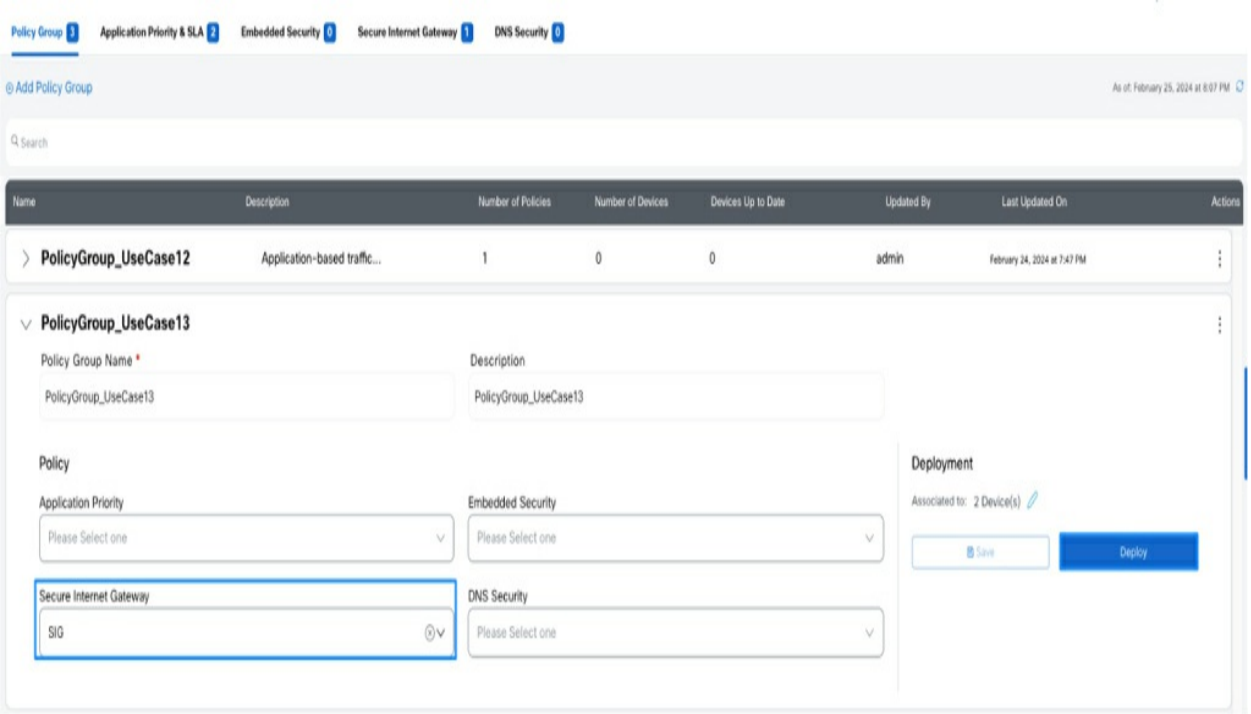


Figure 8-50 *Secure Internet Gateway Policy Deployment*

To complete the configuration, you can add a new sequence to any existing Application Priority & SLA policy associated with the desired WAN Edge devices or create a new policy profile. Figure 8-51 shows Google_Apps also matched and the SIG set to Action so that the traffic is redirected to the SIG provider.

Technet24

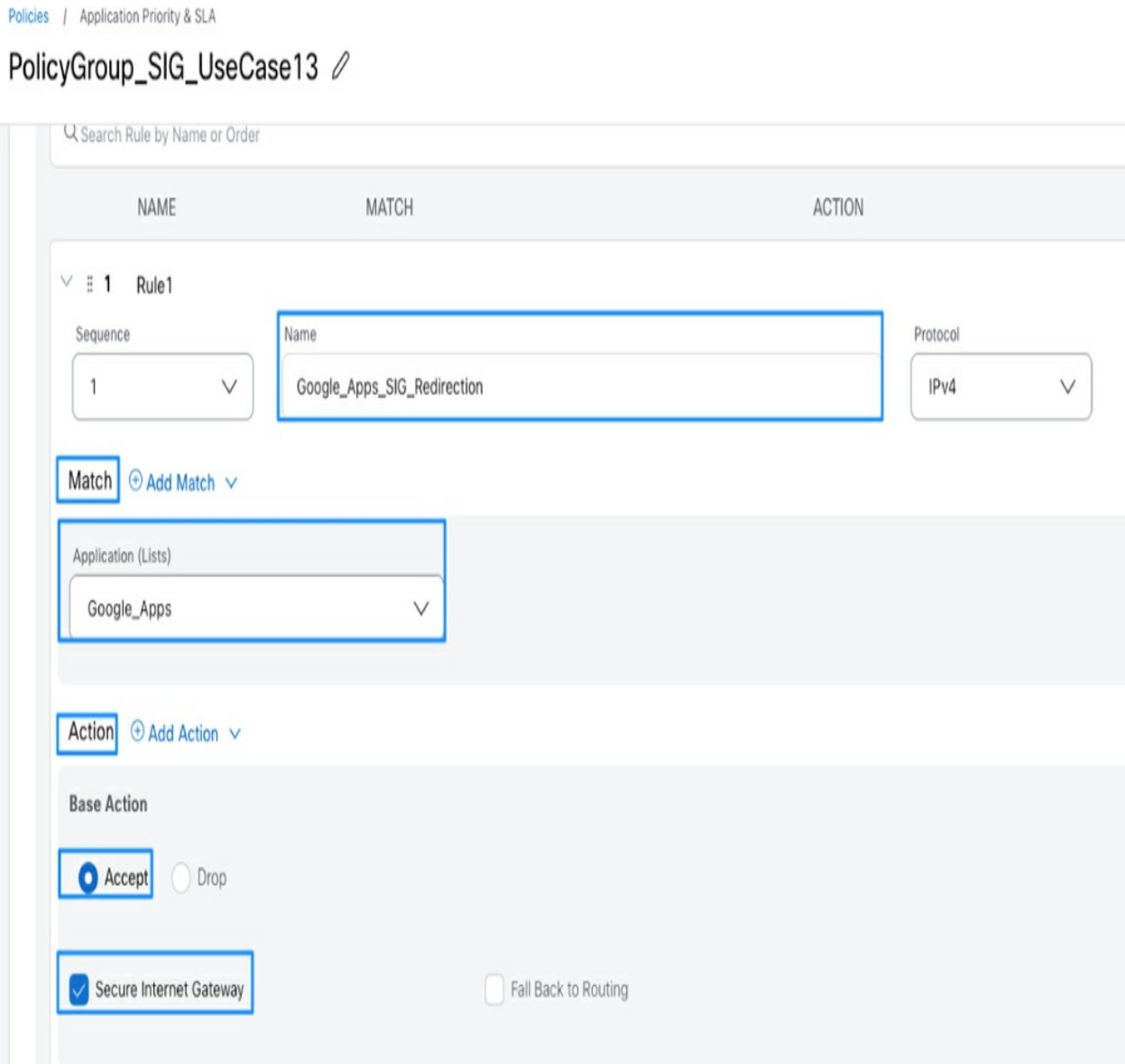


Figure 8-51 *Sequence Rule to Match Google Apps Applications and Redirect Traffic to the SIG Provider*

Finally, append both of the components created—the Application Priority & SLA policy and the SIG policy— to the Policy Group and click Deploy to deploy the changes to the associated devices (see Figure 8-52).

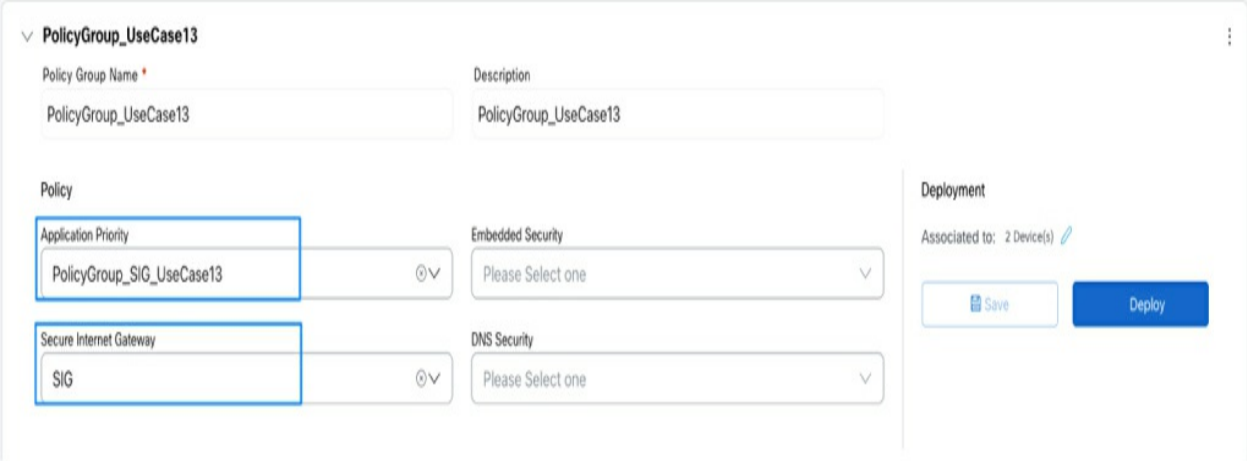


Figure 8-52 *Selecting the Application Priority and SIG Policy Profiles in the Policy Group*

Note

Both Use Case 12 and Use Case 13 demonstrate methods for directing traffic to the Internet by using the **nat use-vpn 0** and **sig** actions. Cloud OnRamp for SaaS provides a more intelligent and dynamic approach to automating the best-path selection for custom and standard applications requiring Internet access. This functionality is detailed in Chapter 12, "Cisco Catalyst SD-WAN Cloud OnRamp."

Use Case 13 Review

This use case shows traditional as well as policy group approaches to redirecting specific traffic flows to Cisco Umbrella SIG. In environments where direct Internet access is leveraged, security and network administrators

Technet24

prefer some set of threat mitigation mechanisms, and SD-WAN Manager makes it possible to execute such mechanisms in a simple way. Chapter 11 provides detailed information about the configuration of these functions.

Use Case 14: Protecting Applications from Packet Loss

This final use case of the chapter covers several tools that can be used to protect applications from the effects of lossy transport links. IP-based networks, by definition, operate at a best-effort level of packet delivery. There is no mechanism at the IP layer to ensure successful delivery of the packet, regardless of whether the underlying transport is a directly connected cable, a service such as MPLS with a contractually guaranteed SLA, or the public Internet. It is typical for all networks to experience some degree of packet loss, and most applications designed to operate on IP networks are engineered to tolerate some degree of packet loss.

One of the most effective ways to counter the effects of packet loss on underlying transport networks is simply to move sensitive applications off transports that are currently experiencing packet loss. In the Cisco SD-WAN fabric, this function is achieved through application-aware routing policies and is discussed in greater detail in Chapter 9, "Application-Aware Routing Policies." However, there are often circumstances (such as when there is only a single transport link available or when all transport links are currently experiencing some degree of packet loss) when simply moving an application off a lossy transport link is not a viable option. In this use case, you can use

two different tools to mitigate the effects of loss: Forward Error Correction and packet duplication.

Forward Error Correction for Audio and Video

Forward Error Correction (FEC) is not a new or novel technology. Different FEC methods have been used with different implementations for decades. The principle behind FEC is that additional information (parity) is transmitted along with the original message so that if a portion of the original message is compromised, the entirety of the original message can be reconstructed. Many commonly used protocols implement some form of FEC at different layers of the networking stack: 40GBASE-T and 100GE Ethernet standards use FEC at Layer 2, and most voice over IP (VoIP) protocols implement some form of FEC at Layer 7. With Catalyst SD-WAN, you can also add FEC to Layer 3 (the network layer).

Figure 8-53 shows the current state of the MPLS transport network on BR2-Edge1. The output indicates that the MPLS network between BR2 and DC1 is currently experiencing 5% to 10% packet loss. This use case shows how to implement FEC in order to mitigate the impact of this lossy transport network on the audio and video applications that the employees are using for collaboration.

Technet24

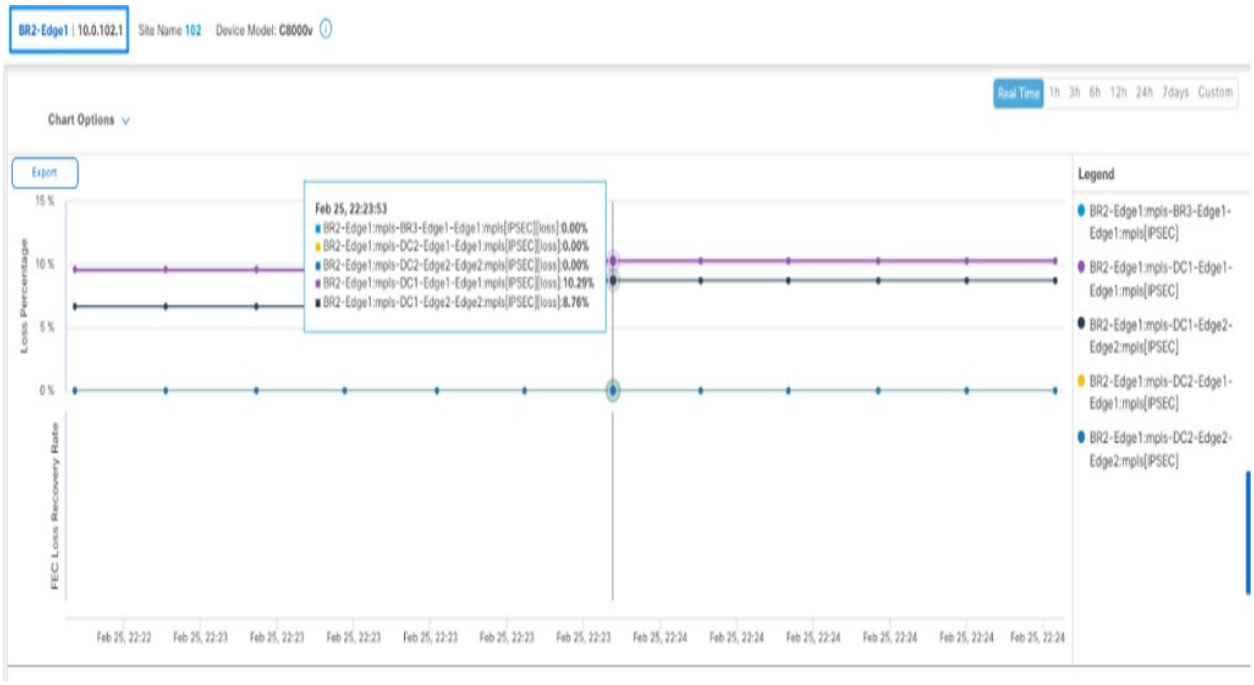


Figure 8-53 *Monitor > Devices > WAN Tunnel Shows the Current Packet Loss Rates*



With Cisco SD-WAN, FEC operates on sets of four packets called a FEC block, as illustrated in Figure 8-54. In the first step, the four packets in an FEC block are processed with a mathematical operation called XOR. The result of this operation is transmitted in the fifth packet, called a parity packet. Each of the packets is encoded with a new FEC header and transmitted to the receiver. If any one of the original four packets is lost in transit to the receiver, as indicated in step 2, but the parity packet is received with the three remaining original packets, the XOR operation is reversed so the lost packet can be reconstructed from the three packets that were received and the parity information that is stored in the fifth packet, as indicated in

step 3. If two or more of the five transmitted packets (four data packets and one parity packet) are corrupted or lost, the lost packets cannot be reconstructed, and only the correctly received packets will be forwarded on to the end host. In this circumstance, the end hosts will notice that packet loss has occurred.

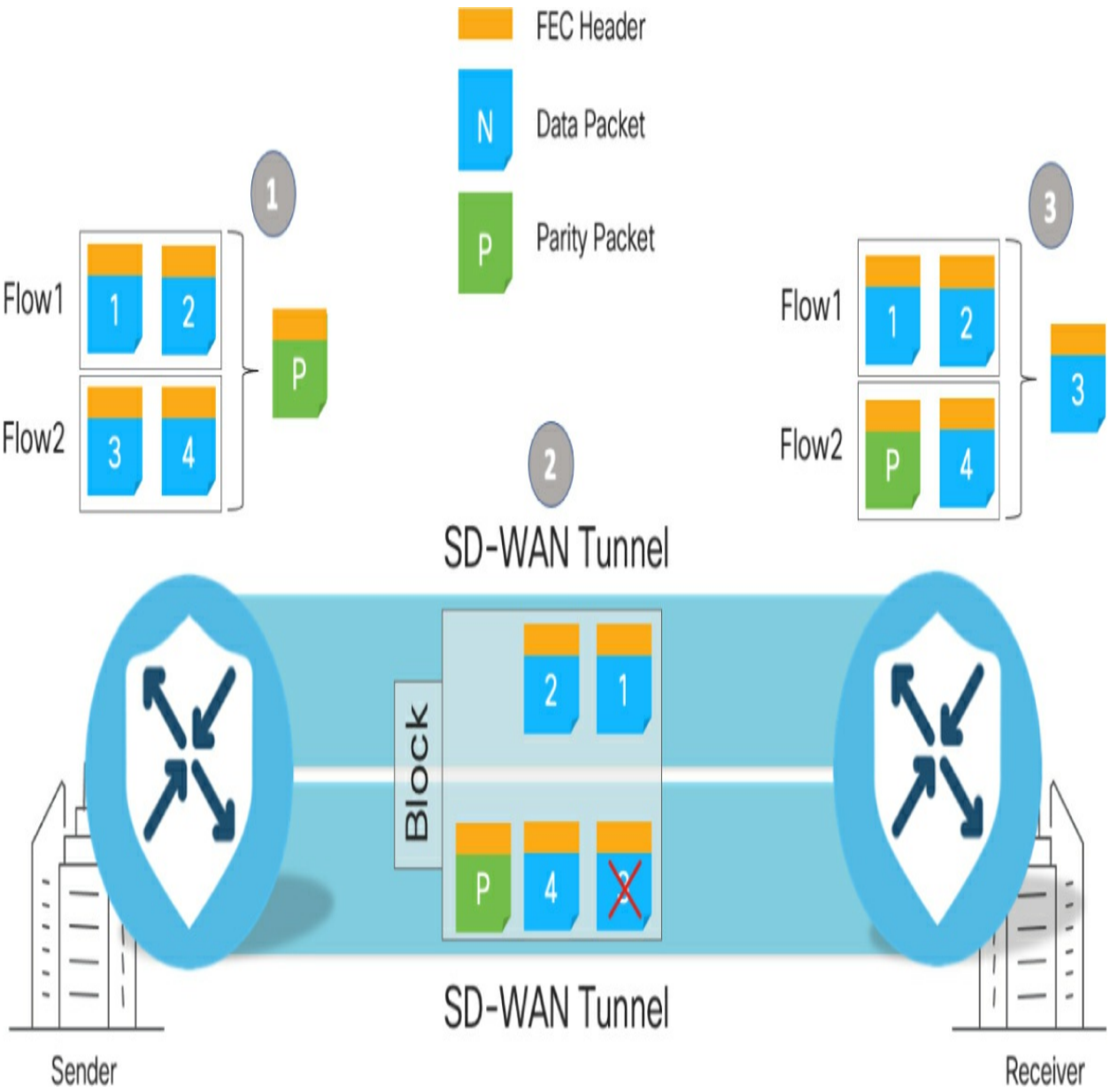


Figure 8-54 *FEC Operation*

Technet24

The process of sending five packets (four data packets and one parity packet) for every four packets of data being transmitted across the WAN would result in an increase of bandwidth consumption of at least 25% (as the parity packet is as large as the largest data packet out of the set of four in the FEC block). While this increase in bandwidth consumption can be beneficial during times of packet loss on the transport links, it may also be unnecessary during a large portion of the time. In order to optimize this, there are two different FEC configuration modes: FEC-always and FEC-adaptive. FEC-always operates exactly as it sounds like it does: The FEC process takes place unconditionally. FEC-adaptive, on the other hand, operates only when the loss percentage on the transport link is detected to be more than 1%. As of SD-WAN Version 20.12, this threshold value can be configured to anywhere between 1% and 5%.

In order to implement this packet loss minimization policy, you need a new sequence to match on the audio/video application family. Then, we need to enable adaptive FEC in order to provide FEC when the transport packet losses are above 1%. In addition to configuring this on the data policy that is applied to the branch routers, you need to configure a corresponding policy on the data center routers so that the traffic that is being forwarded from the data center to the branch sites is also protected by FEC. This policy also needs to configure the **local-tloc** action in order to pin the traffic to the MPLS transport to better illustrate what is happening with the FEC policy. Example 8-12 shows these configurations.

Example 8-12 *FEC Policy*

```
policy
! <<<No changes were made to the control polici
 ! the Guest VPN data policies, and they are or
 !
 ! Branch Data Policy
 !
data-policy _Corporate_Segment_Guest_DIA_FECEna
  vpn-list Corporate_Segment
  ! Sequence 1 matches all of the applications
  ! and turns on fec-adaptive.
  !
    sequence 1
     match
      app-list AUDIO_VIDEO_APPS
      source-ip 0.0.0.0/0
     !
     action accept
      count CORP_AUDIO_VIDEO_-548650615
      loss-protect fec-adaptive
     loss-protection forward-error-correction a
      set
       local-tloc-list
```

Technet24

```
        color mpls
       !
      !
     !
     sequence 11
      match
       destination-data-prefix-list INTERNAL_ADD
      !
      action accept
       count INTERNAL_PCKTS_-548650615
      !
     !
     !<<<remaining sequences are unchanged and c
     !
default-action accept
!
 vpn-list Guest_Segment
 ! <<<Output omitted for brevity>>>
!
! A corresponding policy is also configured on
! to apply the FEC policy to traffic that is b
!
data-policy _Corporate_Segment_Guest_DIA_FECEn
 vpn-list Corporate_Segment
   sequence 1
```

```
     match
      app-list AUDIO_VIDEO_APPS
      source-ip 0.0.0.0/0
     !
     action accept
      count CORP_AUDIO_VIDEO_-431671793
     loss-protect fec-adaptive
     loss-protection forward-error-correction a
      set
       local-tloc-list
        color mpls
      !
     !
    !
 default-action accept
!
lists
! <<<Some lists without changes are omitted fo
!
 app-list AUDIO_VIDEO_APPS
  app-family audio-video
  app-family audio_video
 !
 !
!
```

Technet24

```
!
apply-policy
 site-list Europe_Branches
  control-policy Euro_Reg_Mesh_with_FW_MultiTop
 !
 site-list BranchOffices
 data-policy _Corporate_Segment_Guest_DIA_FECEr
  control-policy Branch_Extranet_Route_Leaking
 vpn-membership vpnMembership_373293275
 !
 site-list DCs
  data-policy _Corporate_Segment_Guest_DIA_FECE
  control-policy DC_Inbound_Control_Policy in
 !
 site-list North_America_Branches
  control-policy North_America_Reg_Mesh_with_FW
 !
!
```

Example 8-12 includes the new sequence that was added to the data policy for the branch sites as well as the new policy that was created and applied to the data center sites. You can see the effects of this policy by using the **show sdwan tunnel statistics fec** command, as shown in Example 8-13.

Example 8-13 *Validating FEC Policy from the Command Line*

```
DC1-Edge1# show sdwan tunnel statistics fec | s
tunnel stats ipsec 192.168.20.1 192.168.20.6 12
 fec-rx-data-pkts     212615
 fec-rx-parity-pkts   52913
 fec-tx-data-pkts     198605
 fec-tx-parity-pkts   49651
 fec-reconstruct-pkts 15496
 fec-capable          true
 fec-dynamic          true
DC1-Edge1#
```

Example 8-13 shows the output from DC1-Edge1, the recipient of the flows from BR2-Edge1. The fec-rx-data-pkts and fec-rx-parity-pkts values are the number of data packets and parity packets, respectively, that have been received by this router. Allowing for slight discrepancies due to the packets that may have been lost in transit, it is notable that approximately 212,615 data packets have been received, which is approximately four times the number of parity packets received (52,913). This is as would be expected since there are four data packets and a single parity packet in each FEC block. The fec-reconstruct-pkts value specifies how many packets were able to be recovered based on the received parity packets. The output indicates

Technet24

that there were 15,496 times when at least three, but not four, of the data packets in the FEC block were received by DC1-Edge1. Using the received parity packet, the router was able to reconstruct the missing packets for these blocks, and the end hosts were unaware that any packet loss occurred during the transmission. The ratio of FEC blocks that were able to be reconstructed to the total number of missing packets can also be seen in the SD-WAN Manager GUI at Monitor > Devices > [Device] > WAN - Tunnel. The fec-tx-data-pkts and fec-tx-parity-pkts values specify how many data packets and parity packets, respectively, have been sent to BR2-Edge1.

Packet Duplication for Credit Card Transactions

As Figure 8-55 indicates, while FEC can be effective at dramatically reducing the number of packets that are lost by the end applications, even in the preceding example, there are many times when the loss recovery rate is not 100%. For circumstances in which the utmost effort needs to be made to have zero packet loss, packet duplication may be the appropriate solution.

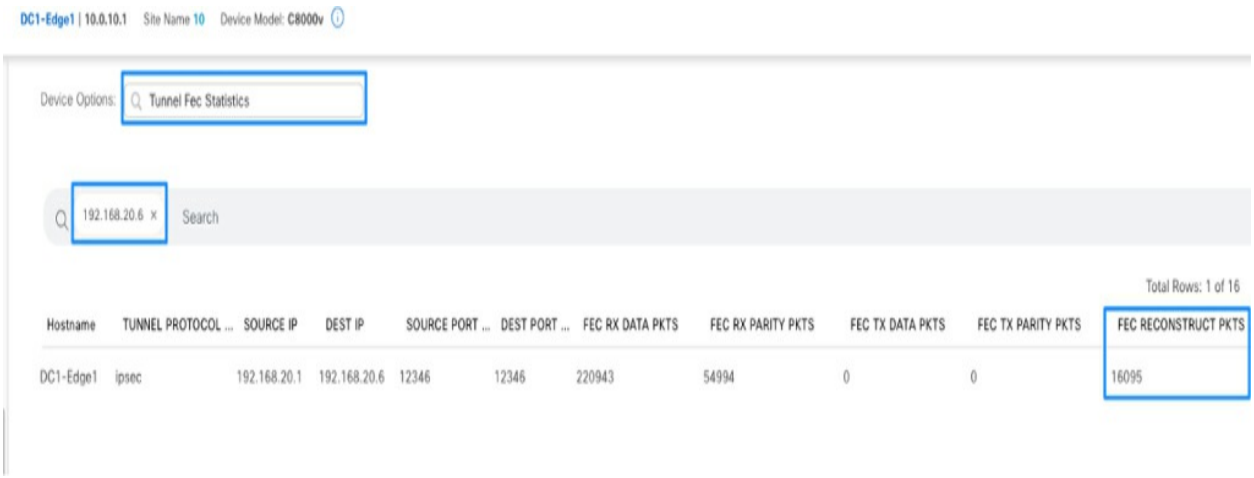


Figure 8-55 *The Current FEC Reconstructs Packets*



Figure 8-56 shows WAN Edge devices forwarding traffic flows with a packet duplication policy enabled. For each packet that is forwarded across a tunnel, a duplicate packet is forwarded across a different tunnel between the same pair of WAN Edge routers. The tunnel that is selected to forward the duplicate packet is the tunnel that currently has the lowest rate of packet loss of any of the tunnels between the pair of WAN Edge devices (excluding the tunnel that was used to forward the original packet).

Technet24

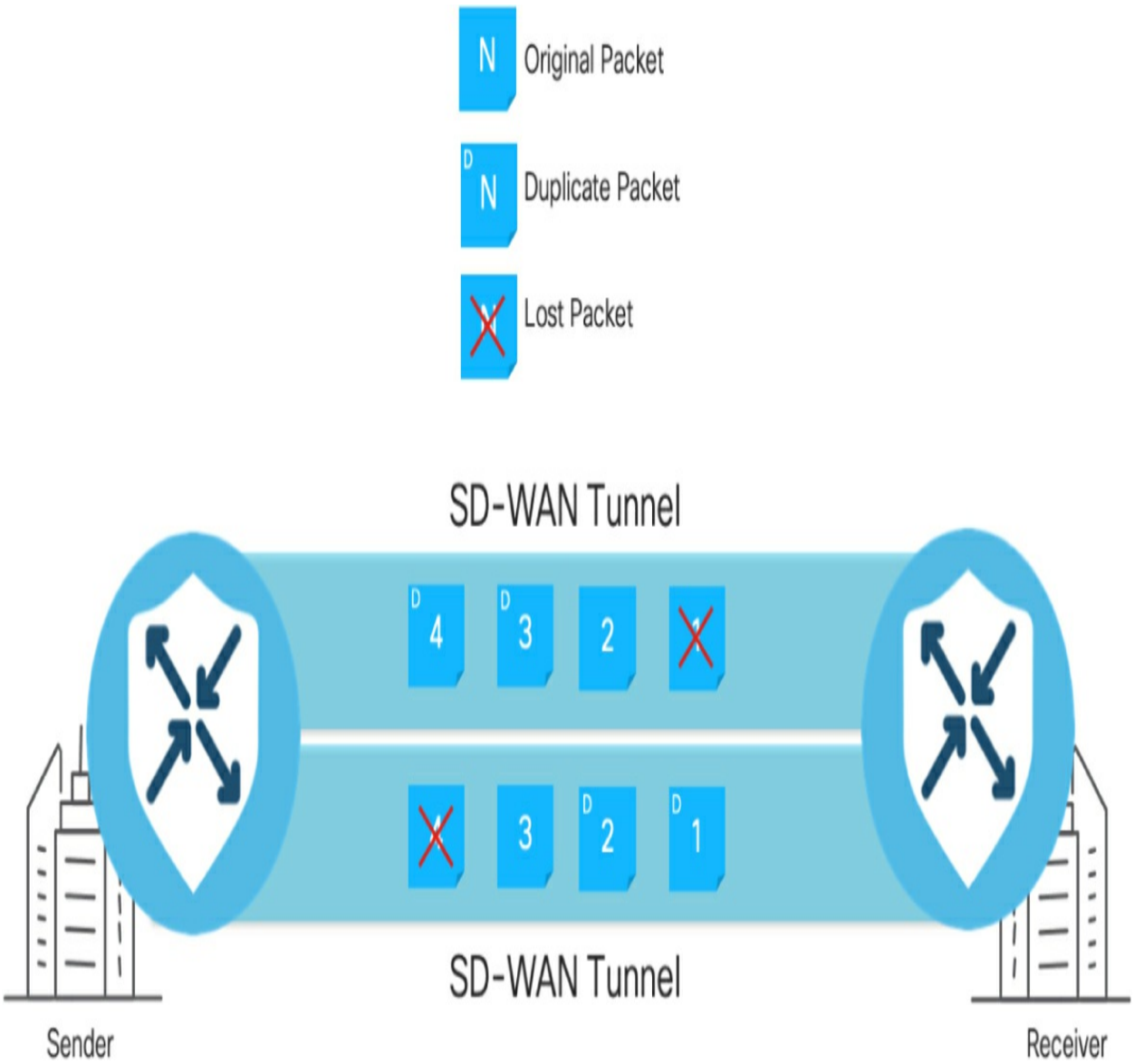


Figure 8-56 *Packet Duplication*

This type of policy is commonly used in retail environments and applied to credit card transactions. Credit card transactions are typically very small flows, but if a packet is lost in transit and is required to be retransmitted, the end-user experience can be slowed down dramatically. For this reason, network administrators find that it may be worth the "price" of transmitting every packet twice in order to ensure that the latency penalty caused by a

packet needing to be retransmitted is avoided.

Example 8-14 demonstrates the forwarding conditions between the credit card reader in Branch 2 and the payment server in the data center through the use of Internet Control Message Protocol (ICMP). The current path between these two sites is experiencing more than 3% packet loss.

Example 8-14 *Packet Loss on the PCI Segment Resulting in Slow Credit Card Processing*

```
C:\Users\BR2-PC> ping 10.20.10.1 -l 10000
Pinging 10.20.10.1 with 10000 bytes of data:
Reply from 10.20.10.1: bytes=10000 time=2ms TTL
Reply from 10.20.10.1: bytes=10000 time=4ms TTL
Request timed out.Request timed out.
Request timed out.Request timed out.

Ping statistics for 10.20.10.1:
    Packets: Sent = 4, Received = 2, Lost = 2 (
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 4ms, Average = 3ms
 C:\Users\BR2-PC>
```

Example 8-15 shows the data policies that are configured on the PCI VPNs in

Technet24

order to perform packet duplication and protect this loss-sensitive traffic. Note that two sequences are configured in both the policy applied to the data center and the policy applied to the branches. One sequence matches traffic with a source address of the payment servers; the other sequence matches the return traffic (traffic with a destination address of the payment servers). Structuring the policy this way enables the policy to be used for both the data center and branch locations simultaneously.

Example 8-15 *Packet Duplication Policy*

```
policy
! <<<No changes were made to the control polici
 ! the Guest VPN or the Corporate VPN data poli
 ! brevity.>>>
 !
 data-policy _Corporate_Segment_Guest_DIA_Packe
  vpn-list Corporate_Segment
  !  <<<Output omitted for brevity>>>
  !
  vpn-list PCI_Segment
    sequence 1
     match
      source-data-prefix-list PAYMENT_SERVERS
     !
```

```
      action accept
       count PCI_PCKTS_-1041354892
       set
        local-tloc-list
         color mpls
       !
       loss-protect pkt-dup
       loss-protection packet-duplication
      !
     !
     sequence 11
      match
       destination-data-prefix-list PAYMENT_SERV
      !
      action accept
       count PCI_PCKTS_-1041354892
       set
        local-tloc-list
         color mpls
       !
       loss-protect pkt-dup
       loss-protection packet-duplication
      !
     !
   default-action accept
```

Technet24

```
!
 vpn-list Guest_Segment
  !  <<<Output omitted for brevity>>>
!
data-policy _Corporate_Segment_Guest_DIA_Packe
vpn-list Corporate_Segment
 !  <<<Output omitted for brevity>>>
!
 vpn-list PCI_Segment
   sequence 1
    match
      source-data-prefix-list PAYMENT_SERVERS
    !
    action accept
     count PCI_PCKTS_-1041354892
     set
      local-tloc-list
       color mpls
      !
       loss-protect pkt-dup
      loss-protection packet-duplication
    !
   !
   sequence 11
    match
```

```
      destination-data-prefix-list PAYMENT_SERV
     !
     action accept
      count PCI_PCKTS_-1041354892
      set
       local-tloc-list
        color mpls
       !
       loss-protect pkt-dup
       loss-protection packet-duplication
     !
    !
  default-action accept
 !
 lists
 ! <<<Some lists without changes are omitted fo
 !
  data-prefix-list PAYMENT_SERVERS
   ip-prefix 10.20.10.0/24
  !
  vpn-list PCI_Segment
   vpn 20
  !
 !
!
```

Technet24

```
apply-policy
 site-list Europe_Branches
  control-policy Euro_Reg_Mesh_with_FW_MultiTop
 !
 site-list BranchOffices
  data-policy _Corporate_Segment_Guest_DIA_Pack
from-service
  control-policy Branch_Extranet_Route_Leaking
  vpn-membership vpnMembership_373293275
 !
 site-list DCs
  data-policy _Corporate_Segment_Guest_DIA_Pack
  control-policy DC_Inbound_Control_Policy in
 !
 site-list North_America_Branches
  control-policy North_America_Reg_Mesh_with_FW
 !
!
```

After the policy in Example 8-15 is applied, the same test is performed again on the credit card reader as was performed in Example 8-14. As Example 8-16 shows, the packet loss has been completely eliminated from the perspective of the end hosts.

Example 8-16 *Results After Packet Duplication Is Applied to the Network*

```
C:\Users\BR2-PC> ping 10.20.10.1 -l 10000
Pinging 10.20.10.1 with 10000 bytes of data:
Reply from 10.20.10.1: bytes=10000 time=2ms TTL
Reply from 10.20.10.1: bytes=10000 time=4ms TTL
Reply from 10.20.10.1: bytes=10000 time=2ms TTL
Reply from 10.20.10.1: bytes=10000 time=4ms TTL

Ping statistics for 10.20.10.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 4ms, Average = 3ms
```

Using the tunnel packet duplication statistics output from the real time display on DC1-Edge1, Figure 8-57 shows in the PKTDUP RX column that only 856 of the original 1074 packets were received on the MPLS tunnel. However, because all 1074 duplicated packets were received on the Internet tunnel, as indicated by the output in the PKTDUP RX OTHER column, the value in the PKTDUP RX THIS column is 1074. The PKTDUP RX THIS column reflects the total number of original packets that the WAN Edge router was able to receive from a combination of the primary tunnel and the tunnel forwarding the duplicate packets.

Technet24

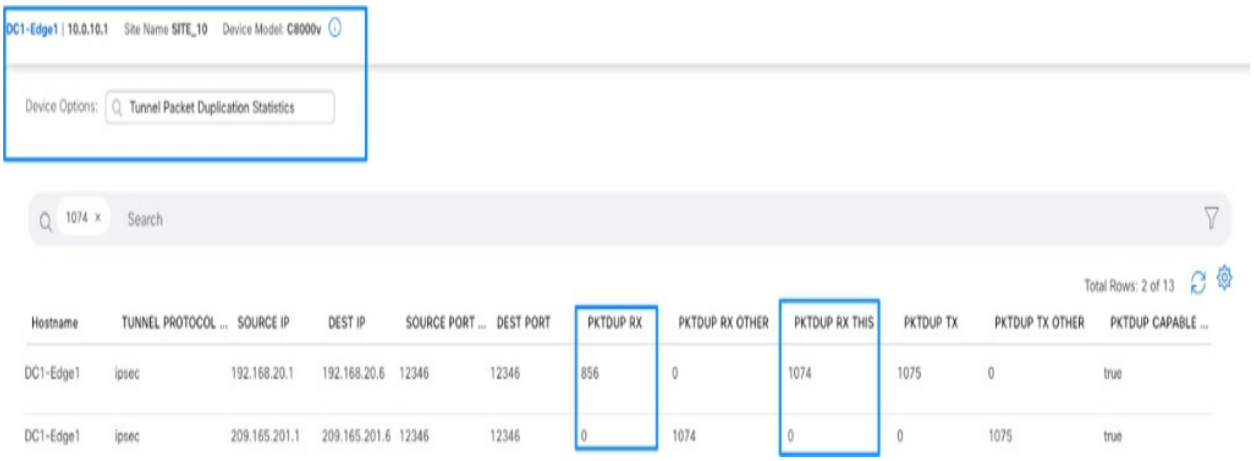


Figure 8-57 *Real-Time Output Showing the Packet Duplication Statistics on DC1-Edge1*

Output in PKTDUP TX and PKTDUP TX OTHER in Figure 8-57 indicates that the 1075 reply packets were also transmitted down the MPLS tunnel and duplicated down the BIZ-INTERNET tunnel. Figure 8-58 shows that packet duplication is doing its job because 1076 packets are being received on both paths. Just as with the DC1-Edge1 router, the 1076 value in PKTDUP RX THIS, which represents the number of original packets received across any transport tunnel and ready to be forwarded to the local end hosts, indicates that no packets were lost in transit for which a duplicate did not successfully arrive.



Figure 8-58 *Real-Time Output Showing Packet Duplication Statistics on BR2-Edge1*



The use case makes extensive use of two different data plane features that are designed to protect against packet loss: packet duplication and Forward Error Correction. Other functionalities, such as TCP optimization (TCP-Opt) and Data Redundancy Elimination (DRE), are part of AppQoE and fundamentally address application performance challenges for data packets as they are forwarded through the router.

Network administrators need to keep prerequisites and minimum requirements in mind when planning and designing their AppQoE deployments, and they should always check the release notes for the latest information about which features are supported on which platforms due to resource requirements. For more information, refer to the "Cisco Catalyst SD-WAN AppQoE Configuration Guide."

Technet24

Use Case 14 Review

This use case explores two different methods for reducing the impact of lossy underlying transport networks on business applications: Forward Error Correction and packet duplication. Chapter 9 discusses another method, Application-Aware Routing, in great detail. Application-Aware routing can also be used to solve similar challenges by moving affected traffic classes off transports that are currently exhibiting packet loss.

Summary

This chapter discusses one of the key types of SD-WAN policies, traditional centralized data policies. It includes several different use cases, focusing on using centralized data policies to make per-application forwarding decisions as well as for direct cloud access, redirecting traffic on a per-application basis toward a Secure Internet Gateway, and mitigating the effects of using lossy transport networks with Forward Error Correction and packet duplication. Some of these use cases look at both the traditional policy approach and using policy groups.

The use cases reviewed in this chapter represent a large cross-section of what centralized data policies can accomplish. There are a plethora of actions not included in the use cases in this chapter, however. Some additional topics that network engineers may be interested in learning more about include quality of service (specifically DSCP marking and remarking), setting the forwarding class on a flow to be matched in a localized policy (for queuing and

scheduling purposes, as discussed in Chapter 10), and using traffic policers. Centralized data policies can also be used to generate Cflowd and NetFlow records that can be exported to an external flow collector for monitoring and reporting purposes. More information and configuration examples for all of these topics can be found in the Cisco documentation.

While these use cases represent some of the most common challenges that network engineers need to solve in order to meet business objectives, the tools and techniques been discussed throughout this chapter can also be applied and extended in novel ways to solve practically any set of requirements.

Review All Key Topics

Review the most important topics in the chapter, noted with the Key Topic icon in the outer margin of the page. Table 8-1 provides a reference for these key topic and the page numbers on which each is found.



Table 8-1 *Key Topics*

Technet24

Key Topic Element	Description	Page
Paragraph, Figure 8-1	Data policy directionality	
Paragraph	Control policies versus data policies enforced on WAN Edge devices	
Paragraph	Concatenating data policies	
Example 8-8	Application traffic engineering behavior with policy in steady state and failed state	
Paragraph	Centralized control policy enforcement behaviors	
Paragraph	Behavior of the **sig** and **nat fallback** policy actions	
Paragraph	Network Wide Path Insight (NWPI) as a tool to gain deep insights into traffic matching in a data policy.	
Paragraph	FEC block operations	
Paragraph	WAN Edge forwarding of traffic flows with a packet duplication policy enabled	
Paragraph	Deployment concerns for the data plane features Forward Error Correction, packet duplication, and APPQOE.	

Define Key Terms

Define the following key terms from this chapter and check your answers in the glossary:

Forward Error Correction

NAT fallback

packet duplication

Chapter Review Questions

1. In a centralized data policy, how do you match all flows?

a. By using the **match-all** criteria

b. By not specifying any matching criteria and only configuring the action statements

c. By using only the default action

2. When the **nat use-vpn** configuration command is used, which VPN is the traffic going to be set to by NAT?

a. VPN 0

b. VPN 1

Technet24

c. VPN 65535

d. The VPN specified in the configuration

3. What is the purpose of the **nat fallback** configuration command?

a. Provides a backup forwarding path in the event that all of the WAN tunnels go down

b. Provides a backup forwarding path in the event that all of the local interfaces configured with NAT go down

c. Provides a backup forwarding path in the event that the destination is not reachable via NAT

4. In an SD-WAN Controller configuration, how many data policies can applied per site?

a. Zero. Data policies are not applied in an SD-WAN Controller configuration.

b. One. Each site gets only one data policy.

c. Two. Each site gets one data policy that is applied to traffic that is originating from the LAN and a second policy that is applied to the traffic that originates from the WAN.

d. As many as necessary but never more than two per VPN per site.

5. True or false: When the TLOC specified in the **local-tloc** action is not available, the traffic that was matched in the data policy sequence is blackholed.

a. True

b. False

6. How many packets are in a single FEC block?

a. One data packet and one parity packet

b. One data packet and four parity packets

c. Two data packets and one parity packet

d. Four data packets and one parity packet

e. The value is configurable in the policy.

7. When adaptive FEC data policies are used, what is the loss threshold at which FEC begins to operate?

a. 0%

b. 1%

Technet24

c. 2%

d. 5%

e. The value is configurable in the policy.

8. When packet duplication is configured, which tunnel is used to send the duplicated packets?

a. The tunnel that is configured in the policy

b. The least utilized tunnel to the same destination

c. The same tunnel that the original packets were sent down

d. The tunnel that is currently experiencing the least amount of packet loss

9. When packet duplication is used, which field indicates the total number of unique packets that have been received?

a. PKTDUP RX

b. PKTDUP RX OTHER

c. PKTDUP RX THIS

d. PKTDUP TX

e. PKTDUP TX OTHER

10. True or false: In order to use policy groups, it is a prerequisite to use configuration groups.

a. True

b. False

11. When redirecting traffic to a secure Internet gateway, what does the action Fallback to Routing mean?

a. Drop the traffic when the tunnels are down.

b. Block the traffic if the tunnels are up.

c. Follow the routing table to find an alternate path.

12. True or false: AppQoE functions such as TCP optimization and Data Redundancy Elimination (DRE) are available through data policies.

a. True

b. False

References

RFC 3871, "Operational Security Requirements for Large Internet Service

Technet24

Provider (ISP) IP Network Infrastructure," https://tools.ietf.org/html/rfc3871.

"Cisco Catalyst SD-WAN Command Reference," https://www.cisco.com/c/en/us/td/docs/routers/sdwan/command/sdwan-cr-book.html.

"Centralized Policy," https://www.cisco.com/c/en/us/td/docs/routers/sdwan/configuration/policies/ic xe-17/policies-book-xe/centralized-policy.html.

"Configure NAT," https://www.cisco.com/c/en/us/td/docs/routers/sdwan/configuration/nat/nat-book-xe-sdwan/configure-nat.html#nat-dia-tracker.

“Policy Groups Configuration Guide,” https://www.cisco.com/c/en/us/td/docs/routers/sdwan/configuration/Policy-Groups/policy-groups/m-policy-groups.html

"Cisco Catalyst SD-WAN Security Configuration Guide," https://www.cisco.com/c/en/us/td/docs/routers/sdwan/configuration/security/ic xe-17/security-book-xe/m-secure-internet-gateway.html.

"Cisco Catalyst SD-WAN AppQoE Configuration Guide," https://www.cisco.com/c/en/us/td/docs/routers/sdwan/configuration/AppQoE/ xe-17/AppQoE-book-xe/m-support-for-multiple-AppQoE-service-nodes.html.

Chapter 9. Application-Aware Routing Policies [This content is currently in development.]

This content is currently in development.

Technet24

Chapter 10. Localized Policies [This content is currently in development.]

This content is currently in development.

Chapter 11. Cisco Catalyst SD-WAN Security [This content is currently in development.]

This content is currently in development.

Technet24

Chapter 12. Cisco Catalyst SD-WAN Cloud onRamp [This content is currently in development.]

This content is currently in development.

Chapter 13. Cisco Catalyst SD-WAN Programmability

This chapter covers the following topics:

• **Cisco Catalyst SD-WAN API Overview:** This section discusses features and capabilities of Cisco Catalyst SD-WAN APIs.

• **Using the Cisco Catalyst SD-WAN API with Python:** This section provides practical examples of how to interact with Cisco Catalyst SD-WAN using REST APIs.

• **Cisco Catalyst SD-WAN Infrastructure as a Code:** This section discusses the infrastructure as code (IaC) approach and its relevance to Cisco SD-WAN deployments.

In the ever-evolving landscape of networking, programmability and automation have emerged as key drivers for agility, efficiency, and innovation, where *network programmability* refers to the ability to control, configure, and automate network devices and infrastructure through software rather than using traditional manual methods. This paradigm shift enables network engineers to dynamically adapt to changing business requirements, deploy new services rapidly, and streamline operations.

Cisco Catalyst SD-WAN is part of this trend. In addition to its foundational capabilities outlined in this book, Cisco Catalyst SD-WAN offers powerful

Technet24

programmability capabilities through its application programming interfaces (APIs). These APIs allow organizations to extend, customize, and automate various aspects of their SD-WAN deployment, unlocking the full potential of their networks for innovation and efficiency.

The adoption of Cisco Catalyst SD-WAN programmability brings several notable benefits:

• **Automation and orchestration:** SD-WAN APIs enable the automation of routine tasks such as provisioning, configuration management, and policy enforcement. By interacting programmatically with the SD-WAN infrastructure, organizations can streamline operations, reduce errors, and speed up service delivery.

• **Customization and extensibility:** SD-WAN APIs provide the flexibility to tailor the network infrastructure to specific business needs. Organizations can develop custom applications, scripts, and integrations that leverage SD-WAN functionality to address unique use cases. This extensibility enables organizations to adapt their network infrastructure to evolving demands and opportunities.

• **Enhanced visibility and control:** SD-WAN APIs offer detailed insights into network performance, traffic patterns, and application behavior. By integrating SD-WAN data with monitoring and analytics platforms, organizations gain actionable insights that enable informed decision making

and proactive network management. In addition, APIs allow organizations to dynamically adjust network policies and configurations in response to changing conditions, ensuring optimal performance and user experience.

• **Accelerated innovation:** SD-WAN programmability stimulates innovation by empowering organizations to experiment, iterate, and innovate rapidly. Developers can use SD-WAN APIs to create new applications, services, and solutions that leverage the capabilities of the underlying network infrastructure. This agility allows organizations to stay ahead of the curve and seize emerging opportunities.

Note

This chapter assumes that you have a basic understanding of programming and APIs, including familiarity with Python programming language, knowledge of JSON data encoding format, and experience with REST APIs. While basic explanations are provided when needed, a comprehensive overview of these topics is beyond the scope of this book. If you are new to network programmability, we recommend checking out the Cisco DevNet program for beginner-friendly resources and guidance.

Cisco Catalyst SD-WAN API Overview

Technet24

Cisco Catalyst SD-WAN uses APIs as a core component of its architecture. Cisco Catalyst SD-WAN APIs (usually called simply SD-WAN APIs) act as a bridge that allows external systems, applications, and services to interact with SD-WAN Manager (formerly vManage), enabling automation, integration, and customization of network management tasks.

Notably, the SD-WAN Manager GUI is an application that interacts programmatically with the SD-WAN Manager servers, much as other clients do. When a user performs actions in the SD-WAN Manager GUI, such as creating a new device template or updating a policy, the GUI application sends corresponding API requests to the SD-WAN Manager server. The server processes these requests, performs the necessary operations on the underlying resources, and returns appropriate responses.

Note

The content in this chapter is based on SD-WAN API Release 20.12, unless otherwise noted. For accuracy, please refer to the SD-WAN API documentation corresponding to the release you are using if you are using a different release.



Cisco Catalyst SD-WAN APIs as REST APIs

SD-WAN APIs adhere to REST principles, allowing clients to use HTTP

requests and standard HTTP methods (such as GET, POST, PUT, and DELETE) to perform operations on resources, which are represented as URLs.

Here's a brief overview of SD-WAN APIs from the REST perspective:

• **Resources:** Each resource in SD-WAN Manager, such as a device, interface, policy, or template, is represented as an API URL. For example, the URL /dataservice/device represents a collection of network devices managed by SD-WAN Manager.

• **HTTP methods:** SD-WAN APIs support standard HTTP methods to perform actions on resources:

• **GET:** Retrieves information about resources.

• **POST:** Creates new resources or initiates actions.

• **PUT:** Updates existing resources.

• **DELETE:** Removes resources.

• **Statelessness:** Each request to SD-WAN APIs is stateless, meaning that the server does not store any client state between requests. Each request contains all the information necessary for the server to fulfill it.

• **Request and response formats:** When requests include the data necessary

Technet24

to perform operations on resources, that data is sent in JSON format in the request body. Responses, also in JSON format, contain data about the operation results.

• **Authentication:** API endpoints are protected by authentication mechanisms to ensure secure access to resources. Clients must authenticate with username and password credentials.

• **Error handling:** SD-WAN APIs provide standard error messages and status codes to indicate the success or failure of requests.

Note

Cisco Catalyst SD-WAN uses two distinct types of APIs: northbound and southbound.

Northbound APIs enable external systems to communicate their intentions to SD-WAN Manager. Cisco Catalyst SD-WAN uses REST APIs as northbound APIs because they provide a flexible, lightweight, and easy-to-use interface.

On the other hand, SD-WAN Manager uses southbound APIs to communicate with network devices. Cisco Catalyst SD-WAN uses NETCONF as a southbound protocol, as it offers a standardized programmatic interface for configuring and managing network devices.

This chapter focuses specifically on northbound REST APIs.

Cisco Catalyst SD-WAN API Guidelines

As mentioned earlier, in an SD-WAN API, each resource is represented as a URL. To illustrate this, let's examine a sample resource URL structure for the API endpoint https://manager.ciscopress.com/dataservice/admin/user. It is constructed from several components, as outlined in Table 13-2.

Table 13-1 *Cisco Catalyst SD-WAN API URL Structure*

Technet24

Component	Description
Protocol	https:// indicates the protocol over which data is exchanged between client and server.
Host:Port	manager.ciscopress.com is the IP address, or hostname, of SD-WAN Manager. The port number may be omitted if the default port, port 443, is used.
API root	/dataservice is a common URL prefix for most SD-WAN API calls. Exceptions include authentication and logout calls.
Resource	The last part denotes the location of the data or object of interest. For instance, the /users resource holds the list of users in SD-WAN Manager.
Parameters	The URL may include optional additional parameters for scoping, filtering, or clarifying a request. In such cases, the question mark (?) starts a query string, and the ampersand (&) separates multiple parameters within a query string. For example, ?deviceId=10.0.10.1 may be used in some API calls to indicate that only data for the device with the system IP address 10.0.10.1 needs to be returned.

Note

In this chapter, we refer to a fictitious SD-WAN deployment that operates with SD-WAN Manager at the https:/manager.ciscopress.com URL. When adapting examples from this book to your specific needs, replace this URL with the address of your organization's SD-WAN Manager.

Example 13-1 presents a sample response to the HTTP GET request sent to the https://manager.ciscopress.com/dataservice/admin/user URL, delivering a list of SD-WAN Manager users in JSON format.

Example 13-1 *SD-WAN API Response Example: A List of Users*

```
{
    "header": {
        "generatedOn": 1710798327336,
        "viewKeys": {
            "uniqueKey": [],
            "preferenceKey": "grid-AdminUser"
        },
        "columns": [
            {
```

Technet24

```
                "title": "Name",
                "property": "description",
                "hideable": false,
                "dataType": "string"
            },
            {
                "title": "Username",
                "property": "userName",
                "dataType": "string"
            },
            {
                "title": "User Groups",
                "property": "group",
                "dataType": "array"
            },
            {
                "title": "Resource Group",
                "property": "resGroupName",
                "dataType": "string"
            },
            {
                "title": "Status",
                "property": "status",
                "dataType": "string"
            }
```

```
        ],
        "fields": [
            {
                "property": "description",
                "dataType": "string"
            },
            {
                "property": "userName",
                "dataType": "string"
            },
            {
                "property": "group",
                "dataType": "array"
            },
            {
                "property": "resGroupName",
                "dataType": "string"
            },
            {
                "property": "status",
                "dataType": "string"
            }
        ]
    },
    "data": [
```

Technet24

```
        {
            "userName": "admin",
            "locale": "en_US",
            "group": []
        },
        {
            "userName": "constantin",
            "description": "Backup local admin
            "locale": "en_US",
            "resGroupName": "global",
            "status": "active",
            "group": [
                "netadmin"
            ]
        }
    ]
}
```

As shown in this example, the standard SD-WAN API response consists of two primary JSON objects:

• **header:** The header object includes a generation timestamp, a columns section, and a fields section. The columns section defines the display name and attributes for the SD-WAN Manager GUI, and the fields section

describes the data type of the response data. This design makes the SD-WAN API self-documenting.

• **data:** The data object contains the actual data retrieved from the SD-WAN system. In Example 13-1, SD-WAN Manager has a default admin user defined along with an additional user constantin, who also has netadmin privileges.

Certain APIs, such as an API that attaches a template to a WAN Edge device, may require substantial processing time. Rather than blocking the caller application until completion, these APIs execute *asynchronously* and just return the task or process ID. The caller application can use this ID to check the state of the request and retrieve the results upon completion. This approach ensures that the caller application remains responsive and can continue with other tasks while awaiting the long-running operation's completion.

Cisco Catalyst SD-WAN API Error Handling

SD-WAN APIs provide standard HTTP status codes, as shown in Table 13-2, to indicate the success or failure of requests. Most HTTP client libraries provide attributes or methods to access the HTTP status code within the response object returned by the API call. For example, with the requests Python library, the status_code attribute of the Response object does this.

Table 13-2 *Cisco Catalyst SD-WAN API HTTP Status Codes*

Technet24

Status Code	Status Message	Meaning
200	OK	Success
201	Created	New resource created
302	Found	Redirect to another URL
400	Bad request	Invalid request
401	Unauthorized	Authentication missing or incorrect
403	Forbidden	Request understood but not allowed
404	Not found	Resource not found
429	Too many requests	Requests exceed rate limit
500	Internal server error	Problem with the server
503	Service Unavailable	Server unable to complete request

In addition, the response body may provide extra information about the

nature of the error. Example 13-2 shows a sample response for a failed API request for the nonexistent device that is returned along with the HTTP 400 status code.

Example 13-2 *SD-WAN API Error Response Example*

```
{
    "error": {
        "message": "Device data error",
        "code": "DEV0001",
        "details": "No device found for system
    }
}
```

When using SD-WAN APIs, be mindful of the following limitations (as of this writing) and make sure your application properly handles rate-limiting and timeout errors:

• There is a limit of 250 concurrent sessions. Because of this limit, it is important to log out at the end of your program.

• Bulk APIs are capped to 48 requests per minute. Bulk APIs allow you to issue a single request to gather information about multiple WAN Edge routers in the overlay network.

Technet24

• Other APIs are restricted to 100 requests per second.

Note

Real-time monitoring APIs are very CPU intensive and should be used only for troubleshooting. They are not suitable for continuous active monitoring of devices.

Cisco Catalyst SD-WAN API Documentation

The official home page for Cisco Catalyst SD-WAN API resources is https://developer.cisco.com/sdwan/. Here, you can find API documentation, "Getting Started" guides, sample code, labs, sandboxes, and more.

Documentation is available at https://developer.cisco.com/docs/sdwan/ (see Figure 13-1). This portal provides API reference and guides for the latest releases of SD-WAN APIs, as well as information for previous releases. (Just click the Latest drop-down to see a list of releases.) Pay attention to Change Logs for SD-WAN API releases, located under the "API Reference" section, as even well-established APIs may change their behavior between releases.

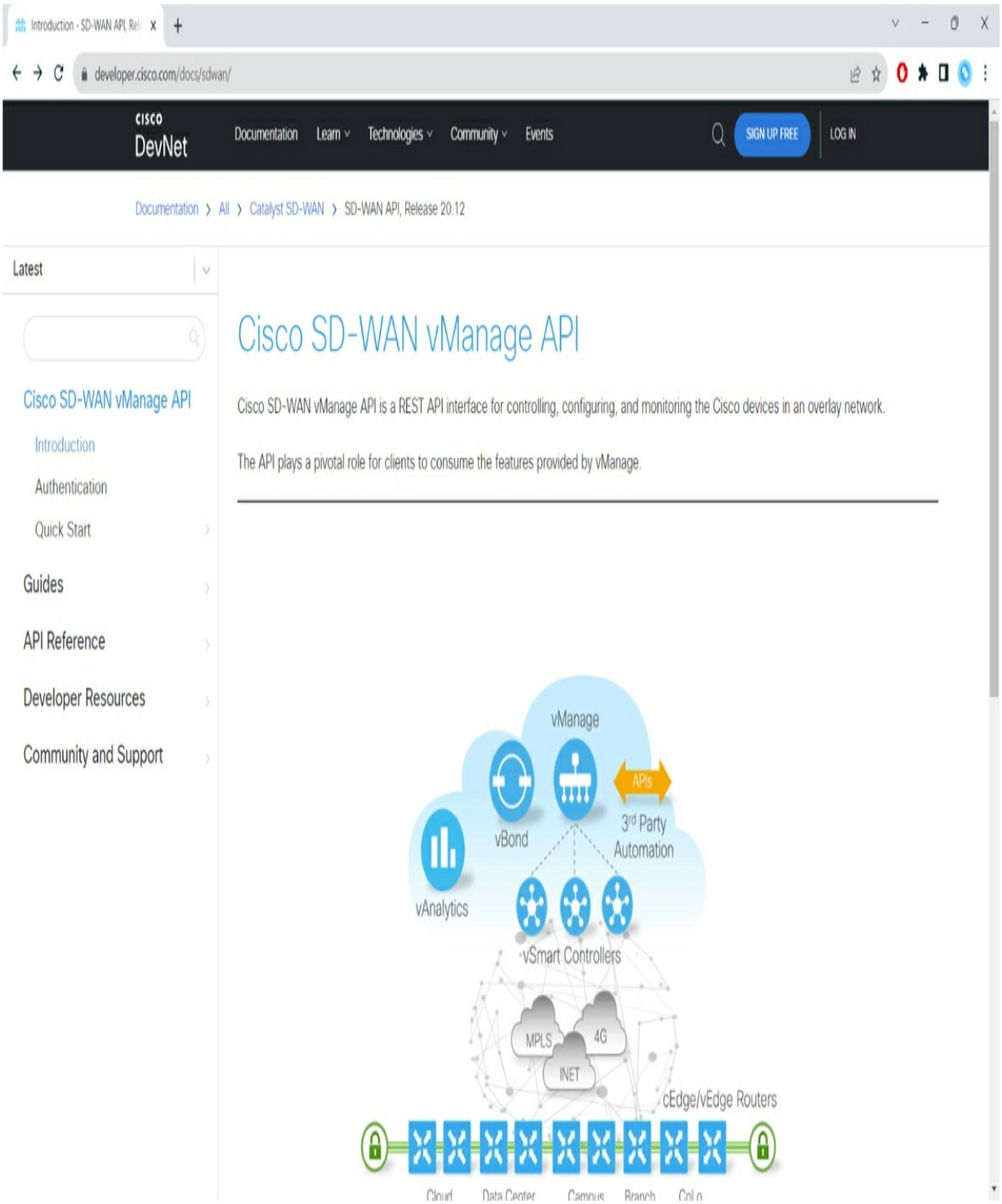


Figure 13-1 *Cisco Catalyst SD-WAN API Documentation Home Page*

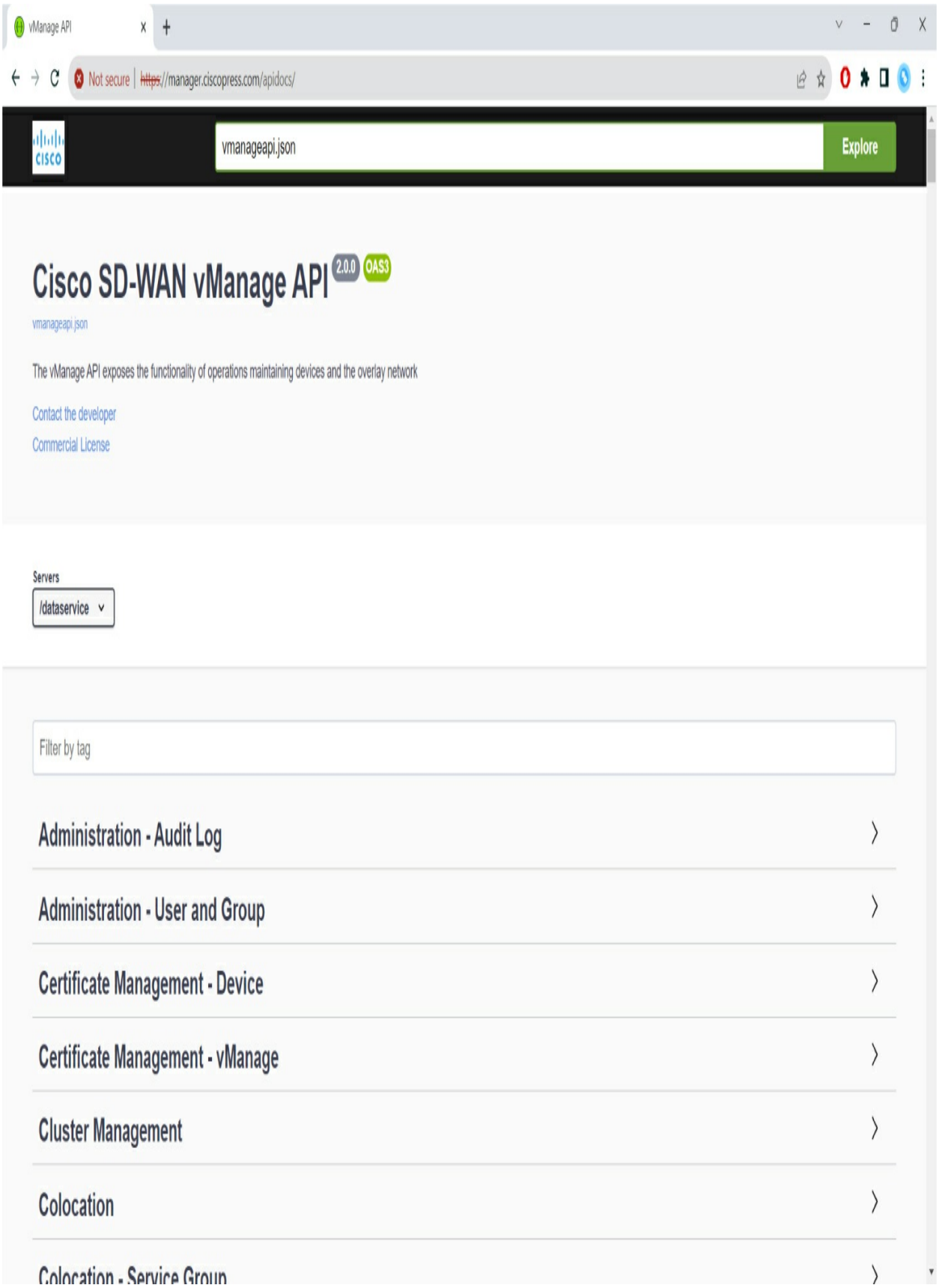
Conveniently, SD-WAN API documentation is also included in the SD-WAN

Technet24

Manager software, accessible at the https://*manager-address*/apidocs (where *manager-address* is the hostname/IP address of the SD-WAN Manager server). On this page, as shown in Figure 13-2, API calls are bundled together into groups for ease of navigation.

Note

As of this writing, there's no link to this URL in the SD-WAN Manager GUI, so you'll have to manually type it into your browser to access the documentation.


vManage API
Not secure | https://manager.ciscopress.com/apidocs/
vmanageapi.json
Explore
Cisco SD-WAN vManage API
2.0.0
OAS3
vmanageapi.json
The vManage API exposes the functionality of operations maintaining devices and the overlay network
Contact the developer
Commercial License
Servers
/dataservice
Filter by tag
Administration - Audit Log
Administration - User and Group
Certificate Management - Device
Certificate Management - vManage
Cluster Management
Colocation
Colocation - Service Group

Technet24

Figure 13-2 *Cisco Catalyst SD-WAN API On-Box Documentation*

Click the arrow on the right to expand a group and list the individual API calls and their respective URL for each call (see Figure 13-3). You can then click on a specific API call title to get detailed information about it, including:

• Description of and notes about the API

• A description of parameters and sample values for them, where applicable

• The request body format and an example (for POST/PUT/DELETE calls)

• Supported response HTTP status codes and examples of responses, where applicable

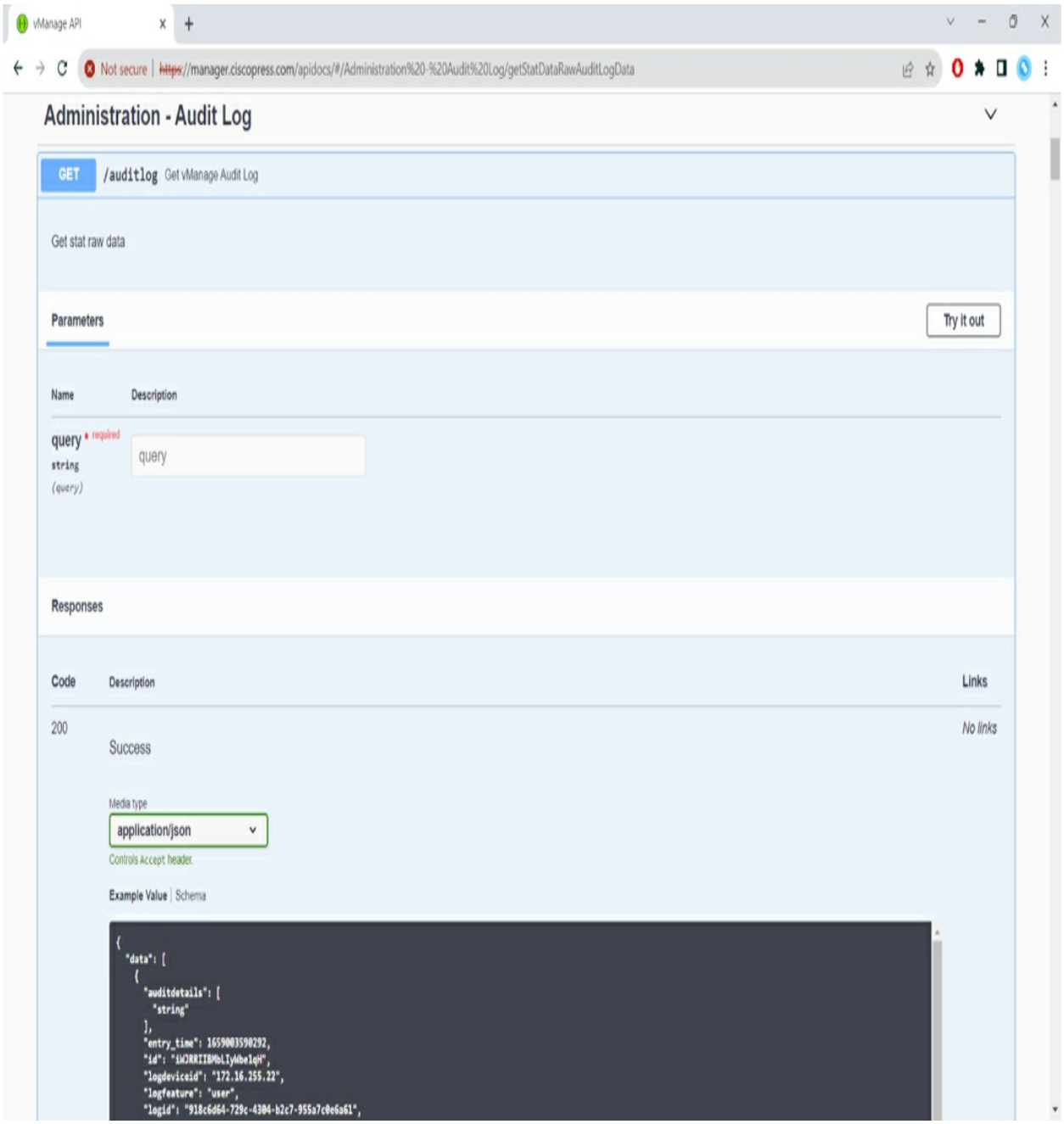


Figure 13-3 *Cisco Catalyst SD-WAN API On-Box Documentation: API Details*

An expanded API page features the "Try it out" button that enables live mode for executing API calls directly on the SD-WAN Manager server. This mode allows you to analyze API behavior interactively, using live data from your

Technet24

SD-WAN fabric. Keep in mind that while GET API calls are generally safe, you should exercise caution when executing PUT, POST, and DELETE API calls due to their potential impact on the system.

Upon providing necessary parameters and clicking the "Execute" button, you are presented with the following information, which is helpful for development efforts:

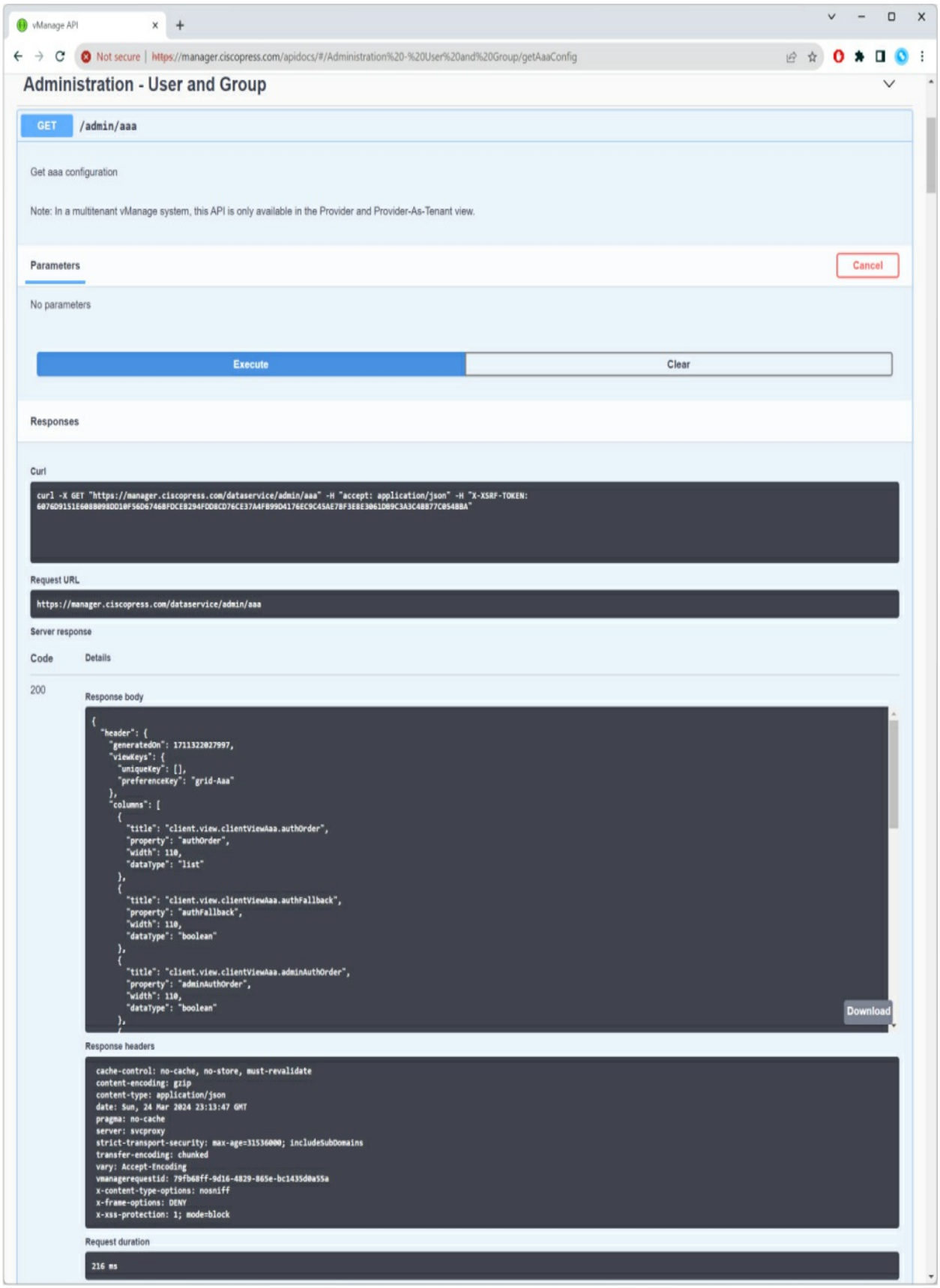
• The command-line equivalent of making an API call with curl (a Linux command-line browsing utility), including all the parameters

• The complete URL of the request

• The SD-WAN Manager server response code, body, and header

• The request execution time

Figure 13-4 provides an example of obtaining a AAA (authentication, authorization, and accounting) setup by making an interactive **GET /admin/aaa** API call. You can find this API in the Administration–User and Group group.


vManage API
Not secure | https://manager.ciscopress.com/apidocs/#/Administration%20-%20User%20and%20Group/getAaaConfig
Administration - User and Group
GET
/admin/aaa
Get aaa configuration
Note: In a multitenant vManage system, this API is only available in the Provider and Provider-As-Tenant view.
Parameters
Cancel
No parameters
Execute
Clear
Responses
Curl
curl -X GET "https://manager.ciscopress.com/dataservice/admin/aaa" -H "accept: application/json" -H "X-XSRF-TOKEN: 6076D9151E6088098DD10F56D6746BFDCE8294FDD8CD76CE37A4FB99D4176EC9C45AE7BF3E8E3061DB9C3A3C4BB77C0548BA"
Request URL
https://manager.ciscopress.com/dataservice/admin/aaa
Server response
Code
Details
200
Response body
{
"header": {
"generatedOn": 1711322027997,
"viewKeys": {
"uniqueKey": [],
"preferenceKey": "grid-Aaa"
},
"columns": [
{
"title": "client.view.clientViewAaa.authOrder",
"property": "authOrder",
"width": 110,
"dataType": "list"
},
{
"title": "client.view.clientViewAaa.authFallback",
"property": "authFallback",
"width": 110,
"dataType": "boolean"
},
{
"title": "client.view.clientViewAaa.adminAuthOrder",
"property": "adminAuthOrder",
"width": 110,
"dataType": "boolean"
},
Download
Response headers
cache-control: no-cache, no-store, must-revalidate
content-encoding: gzip
content-type: application/json
date: Sun, 24 Mar 2024 23:13:47 GMT
pragma: no-cache
server: svcproxy
strict-transport-security: max-age=31536000; includeSubDomains
transfer-encoding: chunked
vary: Accept-Encoding
vmanagerequestid: 79fb68ff-9d16-4829-865e-bc1435d0a55a
x-content-type-options: nosniff
x-frame-options: DENY
x-xss-protection: 1; mode=block
Request duration
216 ms

Technet24

Figure 13-4 *Cisco Catalyst SD-WAN API On-Box Interactive Mode*

Learning from Cisco Catalyst SD-WAN Manager

While the Cisco Catalyst SD-WAN API documentation is comprehensive, navigating through its extensive list of APIs can sometimes be challenging. In addition, the documentation may lack context or overlook certain aspects of API usage. At times, you might want to replicate a specific SD-WAN Manager function programmatically but struggle to find the corresponding API in the documentation. In such situations, observing real-world examples of API usage can be invaluable. And what could be a better source of examples than the Cisco SD-WAN Manager application itself? SD-WAN Manager is just another application that uses an SD-WAN API. What if you could peek into the calls it makes and use them as examples?

Fortunately, modern browsers are equipped with built-in developer tools that allow you to inspect network requests made by web applications. Here's how you can use this feature with the SD-WAN Manager GUI:

Access SD-WAN Manager. Log in to the SD-WAN Manager web interface using your credentials.

Enable developer tools. Most modern browsers include developer tools that you can access by right-clicking on the web page and selecting Inspect or by pressing F12.

Monitor network requests. Within the developer tools, navigate to the Network tab, which shows all network requests made by the web application, including API calls.

Find SD-WAN API calls. Look for requests that are made to URLs starting with /dataservice. These are API calls made by SD-WAN Manager.

Inspect API requests and responses. Clicking on a specific network request provides detailed information about the request and response. You can inspect the request headers, parameters, and payload to understand how the API call is structured. Similarly, you can examine the response body to see the data returned by the API.

Reproduce API calls. Once you have a good understanding of how SD-WAN Manager interacts with the SD-WAN API, you can reproduce these API calls in your own scripts or applications. Use the same or similar HTTP methods, headers, parameters, and payloads to interact with the SD-WAN API programmatically.

Figure 13-5 shows the inspection of the administrative Manage Users page in the Chrome browser. Here you can see that SD-WAN Manager issued an HTTP GET request to https://manager.ciscopress.com/dataservice/admin/user URL to obtain a list of users in the system. The request was executed successfully (according to HTTP code 200 [OK]). You can also see authentication headers, which are explained in the next section.

Technet24

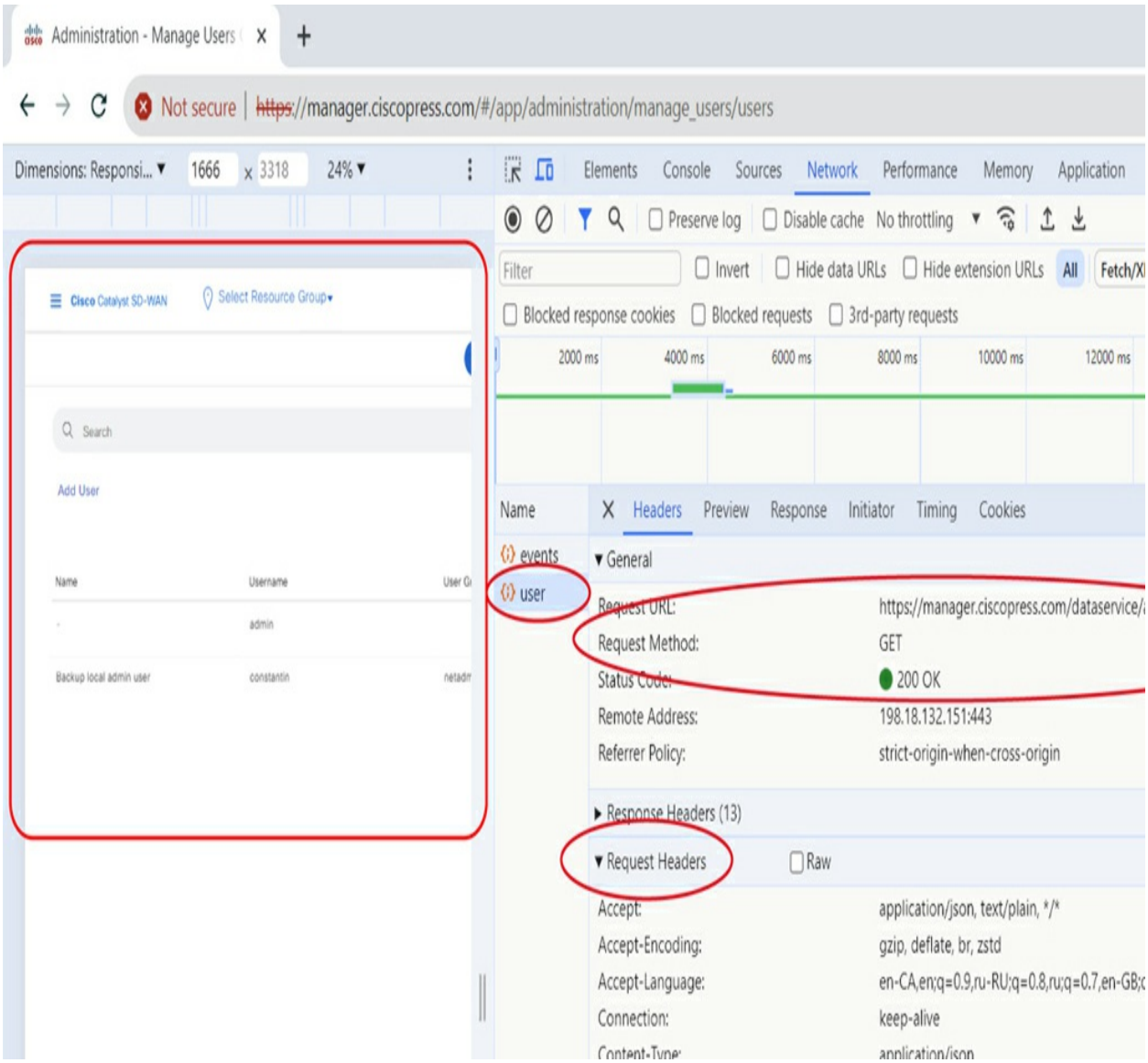


Figure 13-5 *Peeking into SD-WAN Manager's API Use: API Request*

Figure 13-6 shows the response, which is similar to the output provided in Example 13-1 except for the generatedOn timestamp. (Note that the columns and fields sections are collapsed in this figure for the brevity).

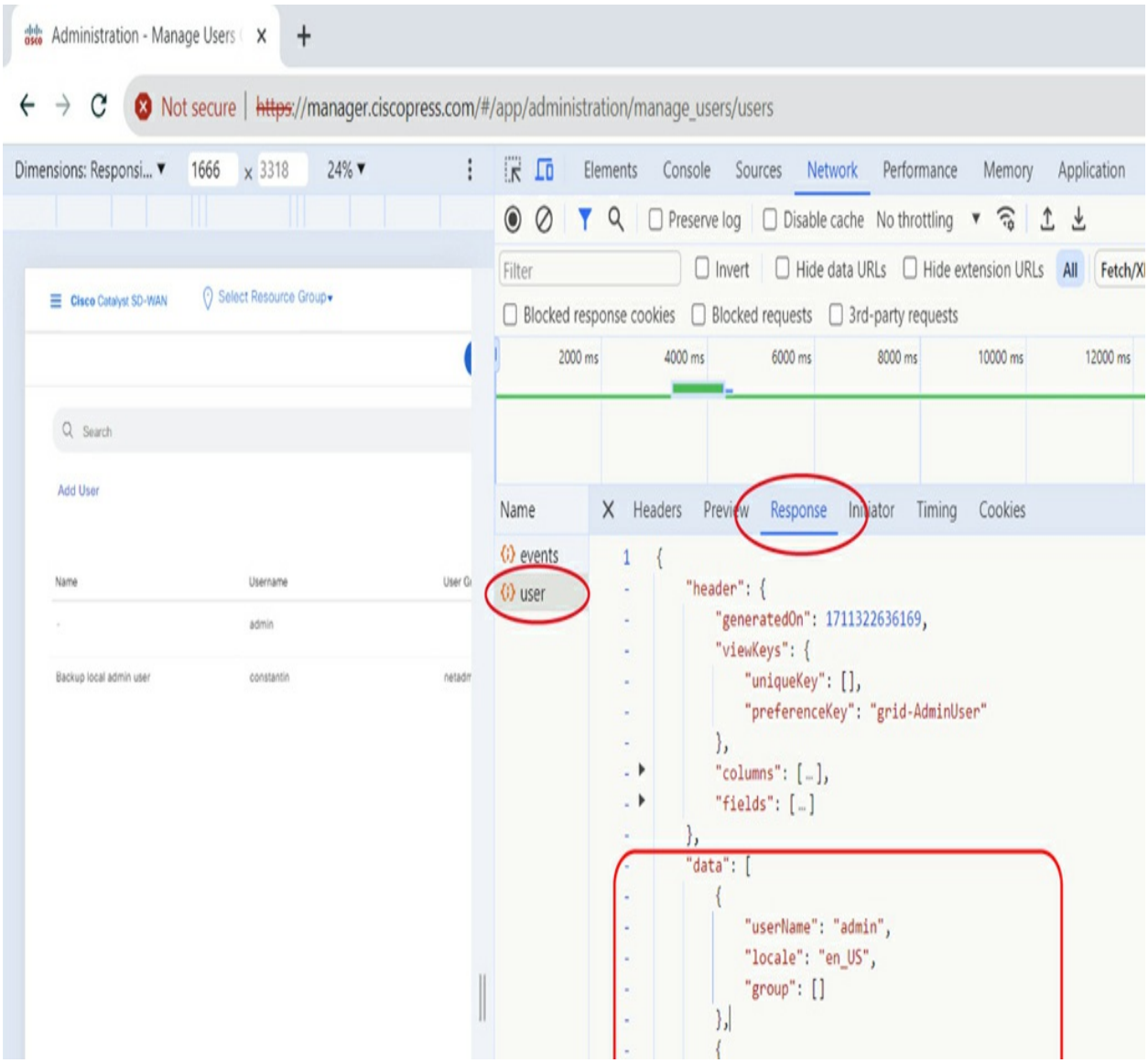


Figure 13-6 *Peeking into SD-WAN Manager's API Use: API Response*

Using the Cisco Catalyst SD-WAN API with Python

This section explores the practical use of SD-WAN APIs, using the Python programming language to illustrate how to interact with different SD-WAN APIs. Python is one of the most popular choices in network programmability due to its simplicity, flexibility, and extensive library ecosystem.

Technet24

A fundamental aspect of working with SD-WAN APIs is the notion of sessions. Upon successful authentication, applications are granted a session that provides them with the privileges needed to interact with the Cisco Catalyst SD-WAN infrastructure. If multiple API requests are invoked sequentially, they normally share the same session. This session-based approach ensures secure access to APIs, with each application being allocated a unique session upon login.

Note

The default lifespan of a session is 24 hours, with an inactivity timeout of 30 minutes. The maximum concurrent session limit is 100. If this limit is reached, the new session will invalidate the least recently used session.

Here's a typical workflow for leveraging an SD-WAN API in a user application:

The user authenticates an application by logging in with their user credentials and establishing a session that grants access to the SD-WAN API endpoints.

An authenticated application can execute API requests based on their logic and requirements. Throughout the session, the application can make multiple API requests, as needed, to accomplish its objectives.

Finally, upon completing its interactions with the SD-WAN API, the application can securely log out and end the session. By terminating the session upon logout, an application revokes access privileges, mitigating the risk of unauthorized access to the SD-WAN infrastructure or other potential security breaches.

Let's review these steps in more detail to gain a better understanding of them.



Cisco Catalyst SD-WAN API Authentication

Before making any SD-WAN API calls, the first step is authentication. The standard method for authenticating with the SD-WAN API is by using username and password credentials with cookie-based authentication. Table 13-3 provides the details of the SD-WAN API authentication call.

Table 13-3 *Cisco Catalyst SD-WAN API Authentication*

Technet24

Parameter	Value/Description
API endpoint	/j_security_check
HTTP Method	POST
Request payload	Username and password in the JSON format: {"j_*username*": *Username*,"j_*password*": *Password*}
Successful call response	HTTP OK (200) response code, with an empty response body and a JSESSIONID session cookie set in the response headers, indicating the allocated session ID
Unsuccessful call response	The same HTTP OK (200) response code but with a response body that contains an HTML login page with an error message

Example 13-3 presents a Python code snippet that performs an authentication call to the SD-WAN API and reports its status along with the session ID upon successful authentication. Note the handling of login credentials: They are either passed via environment variables or prompted for interactively but are not hard-coded. This approach aligns with best practices for working with APIs, ensuring that sensitive information like credentials is not exposed directly in the code.

This and subsequent examples use the requests Python library and its Session object. Using the Session object streamlines interaction with the SD-WAN API by establishing a persistent TCP connection for API requests and automatically handling cookies, headers, and other parameters across multiple requests.

Example 13-3 *SD-WAN API Sample Authentication Code*

```
import requests, getpass, os

# Check environmental variables for SD-WAN Mana
# Use defaults if not defined
MANAGER_ADDR=os.environ.get("MANAGER_ADDR", "ma
MANAGER_USER=os.environ.get("MANAGER_USER") or
MANAGER_PASS=os.environ.get("MANAGER_PASS") or

# API URLs and data structures
base_url = f"https://{MANAGER_ADDR}"
base_api_url = base_url + "/dataservice"

login_url = base_url + "/j_security_check"
login_data = {"j_username": MANAGER_USER, "j_pa

# Helper function, checks and prints status aft
```

Technet24

```
def check_API_status (response):
    print ()
    print (f"Request:  HTTP {response.request.r
    print (f"Response: HTTP code {response.stat

    # Check for critical errors
    response.raise_for_status ()

# Initialize "Session" object from the "request
session = requests.Session ()

# Disable TLS certificate warnings: do it for t
requests.packages.urllib3.disable_warnings()
session.verify = False

# Execute SD-WAN API Authentication call
response = session.post (url = login_url, data
check_API_status (response)

if response.text == "":
    print ("Authenticated successfully, obtaine
    print (f"  {session.cookies['JSESSIONID']}"
else:
    print (f"Authentication failed")
```

Example 13-4 shows the outcome of executing this code. You can see the API call made, the HTTP response code, and the successfully obtained session ID.

Example 13-4 *SD-WAN API Sample Authentication Code Output*

```
Request:  HTTP POST https://manager.ciscopress.
Response: HTTP code 200
Authenticated successfully, obtained session ID
  pOZHwe2Oe9CwhtNte1voqkJGGUM7ivQqr9dAo7uG.e6a8
6cbc27807482
```

With the Session object, the received SD-WAN session ID is automatically stored as a JSESSIONID cookie and does not require any additional handling.

After successful authentication, the application can make read-only (HTTP GET) SD-WAN API calls. However, for write API calls (HTTP POST/PUT/DELETE), an additional authentication element called a cross-site request forgery prevention token (XSRF token) is required. Without it, those APIs will return an HTTP 403 error code along with a token-related error message in the response body. Table 13-4 provides the details of the SD-WAN API call used to obtain the XSRF token.

Technet24

Table 13-4 *Cisco Catalyst SD-WAN API Authentication Token*

Parameter	Value/Description
API endpoint	/dataservice/client/token or /dataservice/client/token?json=true
HTTP method	GET
Request payload	None
Successful call response	Plaintext XSRF token or JSON-formatted token (if called with the json=true option)
Unsuccessful call response	Non-HTTP OK (200) response code

Example 13-5 provides an addition to the sample code from Example 13-3 that obtains the XSRF token. Example 13-6 demonstrates the result of executing this code.

Example 13-5 *SD-WAN API Sample XSRF Token Code*

```
token_url = base_api_url + "/client/token?json=

# Obtain authentication token
response = session.get (url = token_url)
check_API_status (response)

if response.status_code == 200:
    print (f"Token: {response.text}")
    session.headers['X-XSRF-TOKEN'] = response.
else:
    print (f"Unable to obtain the authenticatio
```

Example 13-6 *SD-WAN API Sample XSRF Token Code Output*

```
Request:  HTTP POST https://manager.ciscopress.
Response: HTTP code 200
Authenticated successfully, obtained session ID
  zSX3QB68Pd-1YdQuKFYJfh-zXUVMHbdxhmalMZey.e6a8
6cbc27807482

Request:  HTTP GET
https://manager.ciscopress.com:443/dataservice/
```

Technet24

```
Response: HTTP code 200
Token:
{"token":"CCCFB8CEE5D1A7B18A4D388BCF7A63C90BC4
6FEE07FD4B35B6135EA61326E2318960A"}
```

Having acquired both the session ID and the XSRF token (which is stored for future use as an X-XSRF-TOKEN header in the Session object), the application is now fully authenticated and ready to execute SD-WAN API calls.

Note

To see a practical example of how both the JSESSIONID cookie and the X-XSRF-TOKEN header are used in SD-WAN API calls, refer to Figure 13-5.

Ending a Cisco Catalyst SD-WAN API Session

An application should log out and close its SD-WAN API session once all API activities are complete. This ensures the release of allocated session resources and is considered a best practice for security purposes. Table 13-5 outlines the details of the SD-WAN API logout call.

Table 13-5 *Cisco Catalyst SD-WAN API Logout*

Parameter	Value/Description
API endpoint	/logout?nocache={*random-number*}
HTTP method	POST (for SD-WAN API Version 20.12 and up) or GET (for older SD-WAN API releases)
Successful call response	HTTP redirect (302) code with the location header https://{*manager-address*}/welcome.html?nocache= Note: Your library may handle redirects automatically, as the requests library does by default, so you may see an HTTP OK (200) response code for the redirect target page instead.
Unsuccessful call response	Non-HTTP redirect (302) response code

Example 13-7 extends the sample code from Example 13-5 to include the logout process. Example 13-8 shows the outcome of running this application.

Example 13-7 *SD-WAN API Sample Logout Code*

```
logout_url = base_url + "/logout?nocache="
```

Technet24

```
# Log out
response = session.post (url = logout_url, allo
check_API_status (response)

if response.status_code == 302:
    print (f"Logged out successfully!")
else:
    print (f"Error logging out")
```

Example 13-8 *SD-WAN API Sample Logout Code Output*

```
Request:  HTTP POST https://manager.ciscopress.
Response: HTTP code 200
Authenticated successfully, session ID:
  SPCFLwp6kNpt_kfdMVI8ceeMdlvc6mVkPU_hQvwB.e6a8
6cbc27807482

Request:  HTTP GET
https://manager.ciscopress.com:443/dataservice/
Response: HTTP code 200
Token:
{"token":"C6927A5B9590B57AC826572BE07D94EB8E22D
F04DBF293D84808F54EDB4F2727DE6884"}
```

```
Request:  HTTP POST https://manager.ciscopress.
Response: HTTP code 302
Logged out successfully!
```

This basic application doesn't do much other than login and logout operations. Now let's explore how to use the SD-WAN API to accomplish more practical tasks.



Cisco Catalyst SD-WAN API Categories

Most SD-WAN APIs can be categorized into the following groups:

• **Administration APIs:** These APIs handle tasks related to the management of users, roles, permissions, and systemwide settings in the SD-WAN environment.

• **Certificate management APIs:** These APIs manage digital certificates for authentication and encryption in the SD-WAN fabric, including creation, retrieval, and revocation.

• **Configuration APIs:** These APIs handle various configuration-related tasks in the SD-WAN environment, including manipulation of device settings, inventory, and fabric configuration, as well as handling of templates,

Technet24

configuration groups, and policies.

• **Monitoring APIs:** These APIs provide real-time and historical data on network performance metrics, including traffic statistics, device status, network health, and other metrics for monitoring the SD-WAN environment.

• **Troubleshooting APIs:** These APIs assist in diagnosing and resolving issues in the SD-WAN fabric by providing functionalities such as event log retrieval, path tracing, packet capture, and other diagnostic tools.

The following sections present simple scripts demonstrating practical examples of using these APIs.

Base Python Module for SD-WAN API Interactions

All the basic SD-WAN API interaction tasks will be repeated in each example. To streamline these tasks across all scripts and avoid repetition, let's create a basic Python module that can be reused later. This module uses a Python object to group together data and methods for easier management. This Python object includes authentication and logout functionalities covered in previous sections, along with methods for executing all basic REST calls.

This module, named sdwan_api.py, is presented in Example 13-9. To use it in other scripts, place it in the same directory as a script file and include it with the **import sdwan_api** line in Python.

Example 13-9 *Basic Python Module for SD-WAN API Interactions*

```
import requests, getpass, os, time

class sdwan_api:
  def __init__ (self):

    manager = os.environ.get("MANAGER_ADDR") or
    username = os.environ.get("MANAGER_USER") o
    password = os.environ.get("MANAGER_PASS") o

    self.base_url = f"https://{manager}"
    self.base_api_url = self.base_url + "/datas
    self.session = requests.Session ()

    # Disable TLS certificate warnings: do it f
    requests.packages.urllib3.disable_warnings(
    self.session.verify = False

    if not self.login (username, password):
      raise SystemExit (f'Login to {self.base_u

  def login (self, username, password):
    login_url = self.base_url + "/j_security_ch
    token_url = self.base_api_url + "/client/to
    login_data = {"j_username": username, "j_pa
```

Technet24

```
    response = self.session.post(url=login_url,
                   data=login_data,
                   verify=False)
    response.raise_for_status ()

    # If authenticated, the response body is em
    # If not, the response body contains a html
    if response.text:
      print ("Authentication Error")
      return False

    # session object contains the JSESSIONID co
    response = self.session.get(url=token_url,
    response.raise_for_status ()

    try:
      self.session.headers['X-XSRF-TOKEN'] = re
      return True
    except:
      print ("Error obtaining XSRF Token")
      return False

  def logout (self):
    url = self.base_url + "/logout"
```

```
    # use "GET" action in versions <20.12
    response = self.session.post (url, {'nocach
    response.raise_for_status()

  # basic python method for REST interactions,
  # "method" parameter is HTTP method: GET/POST
  # "url" parameter is full API URL
  # "payload" parameter is payload, where neede
  def api_action (self, method, url, payload =
    try:
      response = self.session.request (method=m
                       json=payload)
      response.raise_for_status()
    except:
      raise SystemExit (f"{method} request fail
                f" code {response.status_code}")

    try:
      data = response.json()
    except:
      data = ""

    # comment to stop debugging:
    print (f"\n>>> HTTP {method} {url}, status
```

Technet24

```
    # uncomment for more debugging:
    # if data:
    #   print (json.dumps(data, indent=4))

    return data

# Python method to execute "GET" REST API cal
def api_GET (self, path):
    response = self.api_action ("GET", self.bas
    return response

# Python method to execute "POST" REST API ca
def api_POST (self, path, payload):
    response = self.api_action ("POST", self.ba
    return response

# Python method to execute "PUT" REST API cal
def api_PUT (self, path, payload):
    response = self.api_action ("PUT", self.bas
    return response

# Python method to execute "DELETE" REST API
def api_DELETE (self, path):
    response = self.api_action ("DELETE", self.
    return response
```

Note

The Python source code for this module, along with all the examples featured in this chapter, is available in the public repository located at https://github.com/sdwanbook/code.

SD-WAN Administrative API Example

The first script demonstrates how to create, read, update, and delete a local user in an SD-WAN system. It uses the following API endpoints, described in more detail in the "Administration–User and Group" section of the API documentation:

• **GET /admin/user:** Reads all local SD-WAN users.

• **POST /admin/user:** Creates a new user based on the provided payload. If the user already exists, an error is returned.

• **PUT /admin/user:** Updates details of an existing user according to the provided payload. If the user does not exist, an error is returned.

• **DELETE /admin/user:** Deletes an existing user from the system.

To illustrate the effect of each of these calls, the script refreshes and prints

Technet24

the list of users after each step. This script, named demo-app-admin.py, is presented in Example 13-10.

Example 13-10 *SD-WAN Administrative API Python Script*

```
import sdwan_api

sample_user = {
    "userName": "demouser",
    "password": "demopassword",
    "description": "Demo User",
    "locale": "en_US",
    "group": ["netadmin"]
}

# Helper function, prints formatted data
def print_users (users):
    for user in users['data']:
        username = user.get('userName','-')
        description = user.get('description','-
        group = user.get('group','-')
        print (f"{username:15}{description:25}-

# ----- start of the script -----
```

```
# Initialize API object
sdwan = sdwan_api.sdwan_api ()

# Print list of current users in the system
print_users (sdwan.api_GET ("/admin/user"))

# Add one more user and print user list to veri
sdwan.api_POST ("/admin/user", sample_user)
print_users (sdwan.api_GET ("/admin/user"))

# Change user group and print user list to veri
sample_user['group'] = ["operator"]
sdwan.api_PUT ("/admin/user/demouser", sample_u
print_users (sdwan.api_GET ("/admin/user"))

# Finally, delete user and print user list to v
sdwan.api_DELETE ("/admin/user/demouser")
print_users (sdwan.api_GET ("/admin/user"))

# Close session and exit
sdwan.logout ()
```

Example 13-11 shows the output of this script.

Technet24

Example 13-11 *SD-WAN Administrative API Python Script Output*

```
>>> HTTP GET https://manager.ciscopress.com/dat
200
admin          -                          []

>>> HTTP POST https://manager.ciscopress.com/da
200

>>> HTTP GET https://manager.ciscopress.com/dat
200
admin          -                          []
demouser       Demo User                  ['netac

>>> HTTP PUT https://manager.ciscopress.com/dat
status: 200

>>> HTTP GET https://manager.ciscopress.com/dat
200
admin          -                          []
demouser       Demo User                  ['opera

>>> HTTP DELETE
https://manager.ciscopress.com/dataservice/admi
```

```
>>> HTTP GET https://manager.ciscopress.com/dat
200
admin           -                          []
```

SD-WAN Device Inventory API Example

The script in this section illustrates how to retrieve inventory information for SD-WAN fabric by using the **GET /system/device/{***deviceCategory***}** API endpoint. The **{***deviceCategory***}** parameter can be either **controllers**, to fetch data about SD-WAN Control Components, or **vedges**, to retrieve data about WAN Edge devices.

By default, this API call returns information about all devices of the specified type. Optionally, you can specify a single device with the deviceIP option (for example, **GET /system/device/vedges?deviceIP=10.0.10.1**). Refer to the API documentation (in the Configuration–Device Inventory section) for this option and additional options.

Note that there is an alternative API call that may be used to obtain the inventory information: **GET /device** endpoint (described in the "Monitoring–Device Details" section of the API documentation). It provides data about provisioned SD-WAN devices, including both SD-WAN Control Components and WAN edge routers. However, this call does not include unprovisioned devices in the response and provides more operational details,

Technet24

such as the number of active BFD sessions, at the expense of configuration details such as the associated template. Choose the API call that better suits your needs.

The script, named demo-app-inventory.py, is presented in Example 13-12. It makes two calls to obtain inventory and displays several elements of the inventory.

Example 13-12 *SD-WAN Device Inventory API Python Script*

```
import sdwan_api

# Helper function, prints formatted data
def print_devices (devices):
    print (f"\n{'Name':12}{'System IP':12}{'Mod
        f"{'Site':5}{'OS Ver':18}{'Template'}")

    for device in devices:
        dev_name = device.get('host-name','-')
        dev_ip = device.get('system-ip','-')
        dev_model = device.get('deviceModel','-
        dev_site = device.get('site-id','-')
        dev_ver = device.get('version','-')
        dev_tpl = device.get('template','-')
```

```
        print (f"{dev_name:12}{dev_ip:12}{dev_r
               f"{dev_site:5}{dev_ver:18}{dev_t

# ----- start of the script -----
# Initialize API object
sdwan = sdwan_api.sdwan_api ()

# Obtain data for controllers
controllers = sdwan.api_GET ("/system/device/co
# Obtain data for routers
wan_edges = sdwan.api_GET ("/system/device/vedg

# print inventory
print_devices (controllers)
print_devices (wan_edges)

# Close session and exit
sdwan.logout ()
```

Example 13-13 presents the output of this script. Here you can observe the API calls made, the HTTP response codes, and inventory information for the SD-WAN fabric., This information includes a list of SD-WAN Control Components and WAN Edge routers, along with the system IP address,

Technet24

device model, site ID, software version, and attached template of each device, where applicable.

Example 13-13 *SD-WAN Device Inventory API Python Script Output*

```
>>> HTTP GET
https://manager.ciscopress.com/dataservice/syst
200

>>> HTTP GET https://manager.ciscopress.com/dat
status: 200

Name         System IP   Model     Site OS Ver
vBond-1      1.1.1.104   cloud     100  20.12.3
vBond-2      1.1.1.105   cloud     100  20.12.3
vManage      1.1.1.101   vmanage   100  20.12.3
vSmart-2     1.1.1.103   vsmart    100  20.12.3
vSmart-1     1.1.1.102   vsmart    100  20.12.3

Name         System IP   Model     Site OS Ver
BR3-Edge1    10.0.103.1  C8000V    103  17.12.0
BR1-Edge2    10.0.101.2  C8000V    101  17.12.0
BR2-Edge1    10.0.102.1  C8000V    102  17.12.0
BR2-Edge2    10.0.102.2  C8000V    102  17.12.0
```

```
BR1-Edge1   10.0.101.1  C8000V    101  17.12.0
DC2-Edge1   10.0.20.1   C8000V    20   17.12.0
DC2-Edge2   10.0.20.2   C8000V    20   17.12.0
DC1-Edge2   10.0.10.2   C8000V    10   17.12.0
DC1-Edge1   10.0.10.1   C8000V    10   17.12.0
-           -           C8000V    -    -
-           -           C8000V    -    -
```

SD-WAN Real-Time Monitoring API Example

The script in this section demonstrates how to obtain operational information for WAN edge devices by using the **GET /device/interface?deviceId=** **{***Device System IP***}** API endpoint (described in more details in the "Real-Time Monitoring–Interface" section of the API documentation). This API call retrieves real-time operational information about interfaces, including state, counters, and other relevant details for a specified device. It provides the same information as if you executed the **show interfaces** command in a router CLI.

While this API call provides a lot of details about the current state of interfaces, the script demo-app-monitoring.py, presented in Example 13-14, displays only a small subset of this information.

Example 13-14 *SD-WAN Real-Time Monitoring API Python Script*

Technet24

```
import sdwan_api

device = "10.0.10.1"

# Helper function, prints formatted data
def print_interfaces (intf):
    print (f"Interface statistics for {intf[0]
            f"({intf[0].get('vdevice-name')})"
    print (f"{'Name':20}{'VPN':6}{'IP Address'
            f"{'Admin':6}{'Oper':6}{'TX octets

    for int in intf:
        if int.get('ip-address') == '0.0.0.0':
            continue
        if_name = int.get('ifname','-')
        if_vpn = int.get('vpn-id','-')
        if_ip = int.get('ip-address','-')
        if_admin = int.get('if-admin-status','
        if_oper = int.get('if-oper-status','-')
        if_tx = int.get('tx-octets','-')
        if_rx = int.get('rx-octets','-')

        print (f"{if_name:20}{if_vpn:6}{if_ip:1
                f"{if_admin:6}{if_oper:6}{if_tx
```

```
# ----- start of the script -----
# Initialize API object
sdwan = sdwan_api.sdwan_api ()

# Obtain real-time interface info
# Real-Time Monitoring - Interface
status = sdwan.api_GET (f"/device/interface?dev

# print interface data
print_interfaces (status)

# Close session and exit
sdwan.logout ()
```

Example 13-15 shows the output of this script.

Example 13-15 *SD-WAN Real-Time Monitoring API Python Script Output*

```
>>> HTTP GET
https://manager.ciscopress.com/dataservice/devi
1, status: 200
Interface statistics for DC1-Edge1 (10.0.10.1)
```

Technet24

```
Name                 VPN   IP Address      Admin
GigabitEthernet1     0     209.165.201.1   up
GigabitEthernet2     0     192.168.20.1    up
GigabitEthernet3     10    172.16.10.1     up
GigabitEthernet4     512   192.168.1.10    up
Loopback65528        65528 192.168.1.1     up
Loopback65529        65529 11.1.10.1       up
Sdwan-system-intf    0     10.0.10.1       up
Tunnel1              0     209.165.201.1   up
Tunnel2              0     192.168.20.1    up
```

SD-WAN Configuration API Example 1

The script in this section can be used for documenting SD-WAN configuration. It lists all the feature templates used by a specified device template. It demonstrates how to retrieve and cross-reference data between multiple API calls, using the following endpoints:

• **GET /template/device:** This API call, described in the "Configuration–Template Master" section of the API documentation, retrieves a list of all configured device templates.

• **GET /template/feature:** This API call, explained in the "Configuration–General Template" section of the API documentation, returns a list of all the configured feature templates.

• **GET /template/device/object/**{*TemplateID*}: This API call, described in the "Configuration–Template Master" section of the API documentation, provides the full content of a specified device template. Example 13-16 shows a sample response.

Example 13-16 *Device Template Sample Response*

```
{
  "templateId": "57dfe544-75d8-40ea-942b-09b63e
  "templateName": "DC1_Edge",
  "templateDescription": "WAN Edge routers in P
  "deviceType": "vedge-C8000V",
[... removed for brevity ...]
  "generalTemplates": [
    {
      "templateId": "aa0fa123-394f-4660-8c9a-b0
      "templateType": "cedge_aaa"
    },
    {
      "templateId": "7a368a9c-e29c-4156-96e3-ec
      "templateType": "cisco_bfd"
    },
[... removed for brevity ...]
```

Technet24

In the script, the first step is to obtain a list of all the configured device templates and feature templates, enabling the mapping of their IDs to names and vice versa. Next, searching by device template name, the script finds the template ID and uses it in the API call to obtain its full content, which includes IDs of all the feature templates used in it. Finally, by mapping feature templates IDs to their names, the script presents the content of the device template in human-readable format.

This script, named demo-app-template-list.py, is presented in Example 13-17.

Example 13-17 *SD-WAN Configuration API Python Script: Example 1*

```
import sdwan_api

target_template = "DC1_Edge"

# Helper function, returns template ID for the
def find_template_id (templates, name):
    for template in templates:
        if template.get('templateName') == name
            return template.get('templateId')
    return None

# Helper function, returns template name for th
```

```
find_template_name (templates, id):
    for template in templates:
        if template.get('templateId') == id:
            return template.get('templateName')
    return None

# Helper function, prints formatted data
def print_device_template (template, feature_te
    print (f"\nDevice Template: {template.get(
    print (f"    Device Type: {template.get('de
','')}")
    print (f"    Description: {template.get('te
    print (f"Attached Feature Templates:")
    # Loop through template's feature templates
    for ftpl in template.get('generalTemplates

        ftpl_name = find_template_name (feature
ftpl.get('templateId'))
        print (f"   {ftpl_name} ({ftpl.get('ter

        # Some templates have sub-templates, ne
        sub_ftpls = ftpl.get('subTemplates')
        if sub_ftpls:
            for sub_ftpl in sub_ftpls:
                sub_ftpl_name = find_template_n
```

Technet24

```
sub_ftpl.get('templateId'))
                print (f"    - {sub_ftpl_name}
({sub_ftpl.get('templateType')})")

# ---------- start of the script ----------
# Initialize API object
sdwan = sdwan_api.sdwan_api ()

# Obtain a list of all device templates
device_templates = sdwan.api_GET ("/template/de
# Obtain a list of all feature templates
feature_templates = sdwan.api_GET ("/template/f

# Find ID - needed for the next API call
target_template_id = find_template_id (device_t
# Obtain content of device template
template = sdwan.api_GET (f"/template/device/ob

print_device_template (template, feature_templa

# Close session and exit
sdwan.logout ()
```

Example 13-18 shows the output of this script. Here, you can see details for

the DC1_Edge device template, such as device type and description, along with a comprehensive list of the feature templates that are included in it.

Example 13-18 *SD-WAN Configuration API Python Script Output: Example 1*

```
>>> HTTP GET https://manager.ciscopress.com/dat
status: 200

>>> HTTP GET
https://manager.ciscopress.com/dataservice/temp
status: 200

>>> HTTP GET
https://manager.ciscopress.com/dataservice/temp
75d8-40ea-942b-09b63ebfc800, status: 200

Device Template: DC1_Edge
    Device Type: C8000V
    Description: WAN Edge routers in Primary Da
Attached Feature Templates:
   Global_AAA (cedge_aaa)
   Default_BFD_Cisco_V01 (cisco_bfd)
   Global_OMP (cisco_omp)
```

Technet24

```
    Default_Security_Cisco_V01 (cisco_security)
    Global_System (cisco_system)
     - Default_Logging_Cisco_V01 (cisco_logging)
    DC1_VPN0 (cisco_vpn)
     - DC1_VPN0_Gig1 (cisco_vpn_interface)
     - DC1_VPN0_Gig2 (cisco_vpn_interface)
    Global_VPN512 (cisco_vpn)
     - Global_VPN512_interface (cisco_vpn_inter
    DC1_VPN10 (cisco_vpn)
     - DC1_BGP (cisco_bgp)
     - DC1_VPN10_Gig3 (cisco_vpn_interface)
    Default_EQUINIX_C8000V_GLOBAL_CISCO_V01 (ce
    Factory_Default_Retail_Banner (cisco_banner)
```

SD-WAN Configuration API Example 2

The advanced script in this example demonstrates how to programmatically modify SD-WAN configuration by adjusting device template variables and pushing updated configuration to attached devices. It uses the following API endpoints:

• **GET /template/device:** This API call retrieves a list of device templates.

• **GET /template/device/config/attached/{***TemplateId***}:** This API call, described in the "Configuration–Device Template" section of the API

documentation, gets a list of devices attached to a specified device template.

• **POST /template/device/config/input:** This API call, explained in the "Configuration–Device Template" section of the API documentation, returns provisioning variables for the template and devices specified in the payload.

• **POST /template/device/config/attachfeature:** This API call, described in the "Configuration–Device Template" section of the API documentation, initiates a task that attaches devices to a template. It does not wait for the task to complete but rather responds with the ID of the created task.

• **GET /device/action/status/{***actionName***}:** This API call, described in the "Configuration–Dashboard Status" section of the API documentation, provides the status of the action, which is the configuration push in this case.

Note

When working in the SD-WAN Manager GUI, attaching a device to a device template and changing device values for an attached device are separate workflows that push configuration to SD-WAN devices. However, from the perspective of the SD-WAN API, both processes—attaching a device to a template and updating device variables—are handled by the same device attachment API call. So don't be confused by the fact that we use the device attachment process when updating device configuration.

Technet24

As in the previous example, the script in this example starts with obtaining the device template ID and identifying devices attached to it. It then retrieves device variables for those devices and updates them as needed. (The interface description is changed to "Configured via API" in this case). Next, the script creates a proper payload and initiates the task to attach devices to a device template, pushing the configuration changes to WAN Edge routers. It will take some time to finish this process, so the script monitors the task's status, checking it every 5 seconds until completion (or timeout). When it is finished, it prints an execution log (returned as a part of the API status call) for every device and the completion status.

This demo-app-template-push.py script is presented in Example 13-19.

Example 13-19 *SD-WAN Configuration API Python Script: Example 2*

```
import sdwan_api, time

target_template = "DC1_Edge"

# Payload structure for the API call that reads
variables_request = {
    'templateId': '',
    'deviceIds': [],
```

```
    'isEdited': False,
    'isMasterEdited': False,
}

# Payload structure for the API call that attac
attach_request = {
  'deviceTemplateList': [
    {
      'templateId': '',
      'device': [],
      'isEdited': False,
      'isMasterEdited': False
    }
  ]
}

# Helper function, returns template ID for the
def find_template_id (templates, name):
    for template in templates:
        if template.get('templateName') == name
            return template.get('templateId')
    return None

# Helper function, checks the status of the asy
# Waits until task completes, with 1 min timeou
```

Technet24

```
def wait_for_task(sdwan, task_id, interval=5, r

    task_url = "/device/action/status/"+task_ic
    time_elapsed = 0
    status = "unknown"

    while True:
        if (time_elapsed < maxtime):
            # read task status from vManage
            status_data = sdwan.api_GET(task_ur

            # current status of operation (in p
            if len(status_data['data']) == 0:
                status = "Validation " +
status_data.get("validation","").get("status","
            else:
                status = status_data['summary']

            # Task is complete when status is "
            if status == "done":
                # Task status includes executio
                print ("\nExecution log:")
                for rtr in status_data.get ('da
                    print (f"{rtr['host-name']]
                    for line in rtr['activity']
```

```
                            print (line)
                        print ()
                    break

            print (f"Operation in progress {tim
{status}")
            time.sleep (interval)
            time_elapsed += interval
        else:
            status = "Timeout"
            break

    return status

# ---------- start of the script ----------
# Initialize API object
sdwan = sdwan_api.sdwan_api ()

# Find template ID - needed for the next API ca
device_templates = sdwan.api_GET("/template/dev
target_template_id = find_template_id(device_te

# Find all the devices attached to this templat
attached_devices =
sdwan.api_GET(f"/template/device/config/attache
```

Technet24

```
']

# Prepare "variables_request" data structure to
print (f"Updating devices attached to the '{tar
variables_request['templateId'] = target_templa
for rtr in attached_devices:
    variables_request['deviceIds'].append (rtr.
    print (f" - {rtr.get('host-name')} ({rtr.ge

# Request device variables
device_variables = sdwan.api_POST("/template/de
variables_request)['data']

# Update values as needed
# In this case, we're updating description of t
for var in device_variables:
    var['/512/vpn512_if_name/interface/descript

# Prepare "attach_request" data structure for t
attach_request['deviceTemplateList'][0]['templa
attach_request['deviceTemplateList'][0]['device

# Template attach - asynchronous call, only ret
task_id = sdwan.api_POST("/template/device/conf
attach_request)['id']
```

```
# Patienly wait for the task to complete
status = wait_for_task(sdwan, task_id)

# Report the status
print(f"Task execution status: {status}")

# Close session and exit
sdwan.logout ()
```

Example 13-20 presents the output of this script. As you can see, the configuration change was successfully applied to two WAN Edge routers attached to the DC1_Edge device template. The process took approximately 15 seconds to complete for both devices.

Example 13-20 *SD-WAN Configuration API Python Script Output: Example 2*

```
Updating devices attached to the 'DC1_Edge' ter
 - DC1-Edge2 (10.0.10.2)
 - DC1-Edge1 (10.0.10.1)
Operation in progress 0/60s, status: Validatior
Operation in progress 5/60s, status: in_progres
```

Technet24

```
Operation in progress 10/60s, status: in_progre

Execution log:
DC1-Edge2, status: Success
[2-Apr-2024 1:26:50 UTC] Configuring device wit
[2-Apr-2024 1:26:50 UTC] Checking and creating
[2-Apr-2024 1:26:52 UTC] Generating configurati
[2-Apr-2024 1:26:55 UTC] Device is online
[2-Apr-2024 1:26:55 UTC] Updating device config
[2-Apr-2024 1:26:55 UTC] Sending configuration
[2-Apr-2024 1:27:00 UTC] Successfully notified
[2-Apr-2024 1:27:02 UTC] Device has pulled the
[2-Apr-2024 1:27:04 UTC] Device: Config applied
[2-Apr-2024 1:27:04 UTC] Template successfully

DC1-Edge1, status: Success
[2-Apr-2024 1:26:50 UTC] Configuring device wit
[2-Apr-2024 1:26:50 UTC] Checking and creating
[2-Apr-2024 1:26:51 UTC] Generating configurati
[2-Apr-2024 1:26:55 UTC] Device is online
[2-Apr-2024 1:26:55 UTC] Updating device config
[2-Apr-2024 1:26:55 UTC] Sending configuration
[2-Apr-2024 1:27:00 UTC] Successfully notified
[2-Apr-2024 1:27:01 UTC] Device has pulled the
[2-Apr-2024 1:27:03 UTC] Device: Config applied
```

```
[2-Apr-2024 1:27:03 UTC] Template successfully
Task execution status: done
```

Sastre Software Development Kit (SDK)

For developers who wish to interact with Cisco SD-WAN without dealing with the low-level details of a REST API and for those who are less familiar with Python, there are other solutions for programmatic interactions — including the Sastre tool.

Sastre is a versatile tool designed to streamline management of configuration elements and visualization of information in Cisco SD-WAN deployments. It can be used as either a self-contained CLI application or as a Python SDK for seamless integration into other applications.

Example 13-21 presents a sample Python application that consumes the Sastre SDK. In this example, the DeviceControlConnections object encapsulates information regarding the state of control connections for the specified device (DC1-Edge1 router). The field_info method provides descriptions of the data fields, and the field_value_iter method facilitates easy iteration over selected data fields.

Example 13-21 *Sastre SDK Python Script*

Technet24

```
import os, getpass
from cisco_sdwan.base.rest_api import Rest
from cisco_sdwan.base.models_vmanage import Dev

device = "10.0.10.1"

manager = os.environ.get("MANAGER_ADDR") or "ma
username = os.environ.get("MANAGER_USER") or in
password = os.environ.get("MANAGER_PASS") or ge

f0,f1,f2,f3,f4='peer_type','system_ip','local_c

with Rest(f"https://{manager}", username, passw
    control_conns = DeviceControlConnections.ge

    # self-documenting API: extract field descr
    head=control_conns.field_info(f0, f1, f2, f
    print(f"{head[0]:11}{head[1]:15}{head[2]:14

    # Iterate through control connections
    for row in control_conns.field_value_iter(f
        print(f"{row.peer_type:11}{row.system_i
              f"{row.local_color:14}{row.remote
```

Example 13-22 presents the output of this script.

Example 13-22 *Sastre SDK Python Script Output*

```
Peer Type  Peer System IP Local Color   Remote
vsmart     1.1.1.102      mpls          biz-int
vsmart     1.1.1.103      mpls          biz-int
vsmart     1.1.1.102      biz-internet  biz-int
vsmart     1.1.1.103      biz-internet  biz-int
vmanage    1.1.1.101      biz-internet  default
```

When Sastre is used as a standalone CLI application, one of its most beneficial features is configuration backup. Sastre can perform a full SD-WAN backup in the API's JSON format, including all the configuration elements, such as templates, policies, and certificates, and it can also save running configurations of all SD-WAN devices in the fabric. This functionality is invaluable for maintaining configuration snapshots and tracking device configuration changes over time.

For more information about Sastre, you can visit the SD-WAN API documentation site at https://developer.cisco.com/docs/sdwan/overview and the official GitHub repository at https://github.com/CiscoDevNet/sastre.

Technet24



Cisco Catalyst SD-WAN Infrastructure as Code

Infrastructure as code (IaC) in networking is a practice that transforms the management and provisioning of network infrastructure resources. Instead of relying on manual configurations through traditional CLI or GUI methods, IaC uses machine-readable configuration files to define, deploy, and manage network configurations, devices, and services as programmable resources.

This approach aligns with the broader DevOps philosophy, fostering collaboration, automation, and agility in network operations. By adopting IaC, organizations can apply software development principles such as version control, automated testing, and continuous integration/continuous deployment (CI/CD) to their network management processes. This shift enables faster, more predictable, and more scalable deployments while reducing manual errors and improving overall efficiency.

In the context of Cisco Catalyst SD-WAN, IaC enables organizations to automate the deployment and management of SD-WAN infrastructure components, including WAN Edge routers, SD-WAN Control Components, policies, templates, and configuration groups. Instead of using manual configurations or direct APIs for each operation, organizations can use tools like Terraform or Ansible to define their SD-WAN infrastructure as code and automate provisioning, configuration, and maintenance tasks.

Terraform and Ansible stand out as the two most popular tools for Cisco Catalyst SD-WAN IaC. Each of them serves a slightly different purpose and takes a distinct approach.

Ansible is often used for configuration management tasks, such as pushing configurations to SD-WAN devices or automating repetitive operational tasks. Some key features of Ansible include:

• Emphasis on simplicity and ease of use

• An imperative approach through Ansible playbooks, which describe actions to execute on target devices (for example, creating the user Alyx Vance and then assigning that user to the group netadmin) rather than the final state of the infrastructure

• A procedural approach in which tasks are executed sequentially on target devices

• Suitability for tasks that require intricate configurations or involve complex logic and conditional statements

Terraform, on the other hand, is typically used for infrastructure provisioning tasks, such as deploying new SD-WAN devices, configuring network connectivity, or deploying security policies. Primary features of Terraform include:

Technet24

• Focus on declarative configuration and state management

• Use of a declarative language to describe the desired infrastructure state (for example, the presence of the Alyx Vance user, a member of the netadmin group), with Terraform managing the provisioning and orchestration of resources to achieve that state

• State management to track the current state of the infrastructure, allowing for easy updates and modifications without disrupting existing resources

• Suitability for provisioning and managing complex infrastructure environments, including multi-cloud setups and hybrid cloud deployments

Table 13-6 highlights the differences between Ansible and Terraform.

Table 13-6 *Ansible and Terraform Comparison*

Feature	Ansible	Terraform
Configuration	Describes actions to execute on target devices	Describes the desired infrastructure state
Approach	Procedural, imperative	Declarative
Language	YAML-based	HashiCorp Configuration Language (HCL)
State management	Manages infrastructure state during runtime	Manages state persistently
Use cases	Configuration management, automation tasks	Infrastructure provisioning, resource management
Strengths	Simplicity, ease of use, complex logic	Declarative configuration, state management

While both Ansible and Terraform support Cisco Catalyst SD-WAN automation, Terraform is often considered superior for IaC due to its declarative approach, infrastructure state management, and broad support for various infrastructure providers. Its ability to manage complex dependencies and orchestrate changes across multiple resources makes it ideal for

Technet24

automating SD-WAN deployments and ensuring consistency and reliability in network configurations.

Ultimately, the choice between Ansible and Terraform depends on the specific requirements of the SD-WAN deployment and the preferred approach to infrastructure automation and management. Some organizations may choose to leverage both tools in combination to capitalize on their respective strengths for different aspects of the SD-WAN environment.

Using Ansible with Cisco Catalyst SD-WAN

For those comfortable working directly with SD-WAN APIs, the simplest way to interact with them in Ansible is by using the built-in uri module, which implements standard interactions with HTTP/HTTPS web servers, as demonstrated in Example 13-23. The output has the same JSON structure as the output from a direct SD-WAN API call.

Example 13-23 *Using an SD-WAN API with the Ansible uri Module*

```
---
- name: Demo Ansible Playbook for API interacti
  hosts: localhost
  gather_facts: no
  vars:
    manager_address:  "{{ lookup('ansible.built
```

```
    manager_username: "{{ lookup('ansible.built
    manager_password: "{{ lookup('ansible.built

  tasks:
  - name: Get Cookie
    uri:
      url: https://{{ manager_address }}/j_secu
      method: POST
      body:
        j_username: "{{ manager_username }}"
        j_password: "{{ manager_password }}"
      body_format: form-urlencoded
      return_content: yes
      validate_certs: no
      status_code: 200
    register: auth_result

  - name: Obtain List of Devices
    uri:
      url: https://{{ manager_address }}/datase
      method: GET
      headers:
        Cookie: "{{ auth_result.set_cookie }}"
      status_code: 200
      return_content: yes
```

Technet24

```
      validate_certs: no
    register: output

  - debug:
      var: "{{ output.content }}.data"
```

Many users prefer to avoid the intricacies involved in manually crafting API calls and instead use specialized Ansible modules tailored for SD-WAN tasks. These modules offer simplified syntax, improved error handling, enhanced maintainability, and reduced overall complexity.

Numerous third-party Ansible modules are available. However, Cisco has developed and provides support for several repositories containing specialized modules that enable users to create specific automation workflows for Cisco Catalyst SD-WAN:

• **ansible-collection-sdwan:** This collection leverages the other two repositories to enable users to create custom workflows or use common ones, such as for full deployment of SD-WAN Control Components and software WAN Edge routers, onboarding configuration, and software upgrades on existing SD-WAN networks. See https://github.com/cisco-open/ansible-collection-sdwan.

• **ansible-collection-sdwan-deployment:** This repository provides the underlying modules needed to deploy and tear down Cisco Catalyst SD-

WAN infrastructure on AWS Cloud. See https://github.com/cisco-open/ansible-collection-sdwan-deployment.

• **ansible-collection-catalystwan:** This repository contains Ansible modules that help users automate SD-WAN tasks like device onboarding and initial configuration, as well as post-bring-up operations. See https://github.com/cisco-open/ansible-collection-catalystwan.

These collections showcase Ansible's modularity and the benefits of using roles, custom modules, and collections for automating network operations. They are valuable resources for organizations seeking to implement infrastructure as code within their networks and adopt a more agile and DevOps-oriented approach to network management.

Example 13-24 illustrates the use of the ansible-collection-catalystwan collection for executing two tasks: health checks and data collection.

Example 13-24 *Using the ansible-collection-catalystwan Ansible Collection*

```
---
- name: Demo Ansible Playbook for cisco.catalys
  hosts: localhost
  gather_facts: false
  vars:
    manager_address:  "{{ lookup('ansible.built
```

Technet24

```
    manager_username: "{{ lookup('ansible.built
    manager_password: "{{ lookup('ansible.built

  tasks:
    - name: "1: Health check: verify all BFD se
      cisco.catalystwan.health_checks:
        check_type: bfd
        filters:
          device_ip: "10.0.10.1"
        manager_authentication:
          url: "{{ manager_address }}"
          username: "{{ manager_username }}"
          password: "{{ manager_password }}"

    - name: "2: Obtain List of Devices"
      cisco.catalystwan.devices_info:
        device_category: vedges
        manager_authentication:
          url: "{{ manager_address }}"
          username: "{{ manager_username }}"
          password: "{{ manager_password }}"
      register: edge_devices

    - debug:
          var: edge_devices.devices
```

SD-WAN Infrastructure as Code with Terraform

Cisco's Terraform SDWAN provider is specifically designed for automating the deployment, configuration, and management of Cisco Catalyst SD-WAN environments with Terraform. Acting as a connector between Terraform and Cisco SD-WAN, this provider allows network administrators to easily define the desired state of their SD-WAN environment, including devices, templates, policies, and configurations—all through Terraform's declarative configuration language.

The Terraform configuration snippet in Example 13-25 shows the basic setup of an SD-WAN environment. It includes some of the SD-WAN configuration resources, such as the feature template resource, the device template resource, and the sdwan_attach_feature_device_template resource that is responsible for keeping per-device variables values and for attaching templates and pushing configuration to WAN Edge devices.

In this example, you can see how Terraform establishes relationship between resources and tracks dependencies between resources. For instance, Terraform understands that a device template cannot be created until its referenced feature templates are created. Furthermore, when a change is made to a resource, Terraform examines dependencies, plans, and executes updates accordingly, including pushing the configuration where necessary, to ensure correct propagation of changes throughout the SD-WAN infrastructure. This

Technet24

dependency-tracking mechanism allows Terraform to efficiently manage and reliably update infrastructure resources in a predictable manner.

This example shows only a single feature template for brevity. We've provided the full configuration in the book's public repository, located at https://github.com/sdwanbook/code, to enable you to explore and implement comprehensive SD-WAN setups using Terraform.

Example 13-25 *SD-WAN Configuration with Terraform*

```
resource "sdwan_cisco_system_feature_template"
  name = "Global_System"
  description = "Global_System"
  device_types = ["vedge-C8000V"]
  console_baud_rate = "9600"
  hostname_variable = "system_host_name"
  site_id_variable = "system_site_id"
  system_ip_variable = "system_system_IP"
}

resource "sdwan_feature_device_template" "DC1_E
  name = "DC1_Edge"
  description = "WAN edge routers in Primary Da
  device_role = "sdwan-edge"
```

```
  device_type = "vedge-C8000V"
  general_templates =   [
    {
      id: sdwan_cisco_system_feature_template.C
      type: sdwan_cisco_system_feature_template
      version: sdwan_cisco_system_feature_templ
    },
..... Output omitted for brevity ........
  ]
}

resource "sdwan_attach_feature_device_template"
  id = sdwan_feature_device_template.DC1_Edge.i
  version = sdwan_feature_device_template.DC1_E
  devices = [
    {
      id = "C8K-PAYG-928-798b-4276-a4b3-2a649a3
      variables = {
        system_host_name = "DC1-Edge1"
        system_site_id = "10"
        system_system_IP = "10.0.10.1"
        vpn0_gig1_ipv4_address = "209.165.201.1
        vpn0_gig2_ipv4_address = "192.168.20.1/
        bgp_neighbor_address = "172.16.10.2"
        bgp_neighbor_remote_as = "65010"
```

Technet24

```
          bgp_router_id = "10.0.10.1"
          vpn10_gig3_ipv4_address = "172.16.10.1/
          vpn512_if_description = "Configured via
          vpn512_if_name = "GigabitEthernet4"
          vpn512_if_ipv4_address = "192.168.1.10/
        },
      },
    ]
  }
```

For further guidance on using the Terraform SDWAN provider, see the comprehensive documentation available at Terraform registry site: https://registry.terraform.io/providers/CiscoDevNet/sdwan/latest/docs.

Summary

This chapter provides a focused overview of network programmability in the context of Cisco Catalyst SD-WAN deployments. It provides an overview of SD-WAN APIs, including guides, error handling, and documentation. The chapter also provides practical API usage scenarios covering authentication, inventory retrieval, monitoring, and configuration management, and provides Python code examples to show practical applications.

This chapter also discusses the infrastructure as code (IaC) approach and its relevance to Cisco SD-WAN deployments. It explores the use of Ansible and

Terraform as powerful tools for automating deployment, configuration, and management tasks in SD-WAN environments. By showcasing real-world examples and code snippets, the chapter equips you with the knowledge and skills necessary to leverage programmability effectively in your SD-WAN infrastructure, promoting efficiency, scalability, and agility in network operations.

Review All Key Topics

Review the most important topics in the chapter, noted with the Key Topic icon in the outer margin of the page. Table 13-7 lists these key topics and the page number on which each is found.



Table 13-7 *Key Topics*

Technet24

Key Topic Element	Description	Page
Section	Cisco Catalyst SD-WAN API as a REST API	
Section	Cisco Catalyst SD-WAN API authentication	
Section	Cisco Catalyst SD-WAN API categories	
Section	Cisco Catalyst SD-WAN infrastructure as code	

Key Terms

Define the following key terms from this chapter and check your answers in the glossary:

Ansible

infrastructure as code (IaC)

network programmability

REST API

Terraform

Chapter Review Questions

1. What kind of APIs are Cisco Catalyst SD-WAN APIs?

a. SOAP

b. REST

c. RPC

d. WORK

2. What communication protocol would you use to interact with Cisco Catalyst SD-WAN APIs?

a. Telnet

b. SSH

c. HTTP

d. WORK

3. When interacting with SD-WAN APIs, which HTTP methods can be used? (Choose two.)

Technet24

a. GET

b. PULL

c. PUSH

d. PUT

4. Which of the following is a valid SD-WAN API URL?

a. https://manager.ciscopress.com/data/admin/user

b. http://manager.ciscopress.com/dataservice/admin/user

c. https://manager.ciscopress.com/dataservice/admin/user

d. http://manager.ciscopress.com/data/service/admin/user

5. Which HTTP response code indicates a successful SD-WAN API request?

a. 201

b. 302

c. 403

d. 503

6. True or false: The SD-WAN Manager GUI is a standalone web application

that uses SD-WAN APIs to perform its functions.

a. True

b. False

7. Which of the following is not a valid SD-WAN API category?

a. Certificate management

b. Configuration

c. Monitoring

d. Assurance

8. True or false: An SD-WAN API call to attach a template to a WAN Edge device returns the status of the configuration push after its completion.

a. True

b. False

9. How is network configuration managed with the infrastructure as code (IaC) approach?

a. Via the CLI

Technet24

b. Via the GUI

c. Via machine-readable configuration files

d. Via PowerPoint slides

10. What is true about IaC tools?

a. Ansible configuration typically defines the final state of the infrastructure.

b. Ansible uses HCL as the configuration language.

c. Terraform does not track the state of the infrastructure.

d. Terraform uses declarative language to describe the desired state.

References

Public repository for the code used in this chapter: https://github.com/sdwanbook/code.

“Cisco Catalyst SD-WAN API,” https://developer.cisco.com/sdwan/.

“Cisco Catalyst SD-WAN Manager API,” https://developer.cisco.com/docs/sdwan/.

Sastre tool repository: https://github.com/CiscoDevNet/sastre.

“SDWAN Provider,” https://registry.terraform.io/providers/CiscoDevNet/sdwan/latest/docs.

Technet24

Chapter 14. Cisco Catalyst SD-WAN Monitoring and Operations [This content is currently in development.]

This content is currently in development.

Glossary of Key Terms

A

Ansible An open-source automation and configuration management tool that uses easily readable playbooks to automate configuration changes while ensuring that the desired state is met.

app-probe-class A setting that includes forwarding class, color, and DSCP values and that defines the marking, by color, of the applications that are forwarded.

app-route multiplier A setting that determines how many app-route poll intervals should be considered when making a determination about the SLA compliance of the tunnels. The default value is 6, and the maximum is 6. This value is configured per router.

app-route poll interval A setting that defines the period of time to collect Bidirectional Forwarding Detection (BFD) probes for analyzing the statistical performance of SD-WAN tunnels and making a determination about SLA compliance. This value is configured per router.

application programming interface (API) A flexible interface beyond the traditional user interface that can be used programmatically to manage and monitor an application, a device, or an operating system.

Technet24

artificial intelligence (AI) The use of compute power to make human-like and informed decisions based on real-time data in the environment.

B

backup-sla-preferred-color An optional configuration argument that allows for the specification of a selected color or colors to use when forwarding a class of traffic in the event that no tunnels meet the required SLAs.

BFD hello interval A setting that specifies how often a WAN edge router should send a BFD probe on a tunnel. This value is configured per router, per color.

BFD multiplier A setting that specifies how many consecutive BFD probes can be sent without a response before the tunnel is declared to be down. This value is configured per router, per color.

bring your own device (BYOD) A common enterprise administrative policy that allows for employees to connect to enterprise networks or the Internet with personal devices such as phones and tablets.

C

centralized policy A policy that can affect the entire Cisco SD-WAN fabric and is activated on SD-WAN Controller.

certificate authority (CA) An entity that is responsible for signing certificate

requests and issuing SSL certificates. Since SD-WAN components are configured to trust an organization's root CA, any certificate generated or signed by the root CA is also trusted. Hence, SD-WAN components inherently trust the identity of one another since they share the same mutual trust of the signing root CA.

Cisco Catalyst Software-Defined WAN (Cisco SD-WAN) A software controller–based solution from Cisco that uses SDN to deploy, monitor, and manage wide-area networks.

cloud Shared compute and application resources that exist in a domain away from the physical enterprise network, such as the Internet or a shared data center. Examples include Amazon Web Services (AWS), Google Compute (GCP) and Microsoft Azure.

color An attribute that allows an SD-WAN solution to identify specific transports and influence how the data plane is built.

command-line interface (CLI) A tool for configuring network devices individually by inputting configuration commands.

control plane (SD-WAN Controller) The element of SD-WAN where all control and centralized policy are enforced. Calculation of the routing table and distribution of encryption keys are handled by the SD-WAN Controller in the control plane.

Technet24

control policy A policy that manipulates routing information and can be used to affect how traffic is forwarded through a WAN edge device.

D

data plane (WAN edge) The element of SD-WAN where data traffic is terminated and encapsulated across the SD-WAN fabric. The data plane is only built between WAN edge devices.

data policy A policy that directly impacts the forwarding of traffic flows through a WAN edge router.

Direct Cloud Access (DCA) Cisco technology that forwards SaaS traffic (such as Office 365, Salesforce, Box, and Google traffic) from a branch directly to the Internet or the backhaul path to a data center based on candidate path performance. It ensures the best SaaS application experience and also reduces the IT WAN cost.

Direct Internet Access (DIA) Technology that makes it possible to access the Internet through local egress at the remote site rather than backhauling through a data center.

E–I

enhanced application-aware routing (EAAR) A type of routing that has the capability to read into the application data traffic which allows for more

accurate and detailed measurements of these metrics.

extranet A restricted communications network that is typically used to allow business partners in different organizations to have a private and secure communication channel.

fallback-to-best-path An option that selects the best path out of all the available colors when an SLA isn't met. When it is selected, you can choose more options that determine the best color.

Forward Error Correction (FEC) A method of including additional information, called parity, in a message so that if part of the message is lost or corrupted, the whole message can still be recovered.

inbound control policy A control policy that is applied to OMP updates sent from a WAN edge device to a SD-WAN Controller and applied before the SD-WAN Controller performs the best-path selection algorithm.

infrastructure as code (IaC) The ability to define the desired configuration as human-readable code. You can use this code with automated DevOps practices rather than manual operations.

infrastructure as a service (IaaS) Virtualized hardware that is outsourced to providers and that typically runs in the cloud.

Internet of Things (IoT) A collection of nontraditional network-connected

Technet24

devices that are typically unstaffed, such as manufacturing equipment, lighting, security cameras, and door locks.

intrusion detection system (IDS) A system that analyzes network traffic for signatures that match known cyberattacks.

intrusion prevention system (IPS) A system that analyzes packets and that can also stop the packets from being delivered based on the kind of attack that is detected, thus helping stop the attack.

L–M

localized policy A policy that affects only a single WAN edge router and that is configured in the device template.

machine learning A subset of AI that is used to gather data and information from the network environment to constantly learn, adapt, and improve the accuracy of the AI.

management plane (SD-WAN Manager) The element of Cisco Catalyst SD-WAN where day-to-day administration of the SD-WAN fabric occurs. SD-WAN Manager is a single pane of glass used for configuration, troubleshooting, software upgrades, and monitoring.

multi-topology A network design in which different VPN segments have different logical topologies. Some VPNs may be able to establish direct

communication with each other, whereas other VPN segments may have to communicate indirectly via a third site, and other VPN segments may not be able to communicate at all.

N–O

NAT fallback The process of forwarding traffic that would have been subjected to network address translation through a local egress interface across the SD-WAN fabric when no local interfaces are configured for NAT in an operational state.

network programmability The use of code to write programs or processes that communicate with network infrastructure. Various programming languages support network programmability, and the most common of them are Python and Go.

OMP route A route that is responsible for carrying information about data prefixes. Such a route is usually a LAN subnet.

orchestration plane (SD-WAN Validator) The glue that brings together all the other SD-WAN Control Components. The orchestration plane distributes SD-WAN Manager and SD-WAN Controller information to the WAN edge devices and also authenticates all the Cisco SD-WAN Control Components.

originator A matching criterion that is used to select the WAN edge device that did the initial advertisement of a route or TLOC route.

Technet24

outbound control policy A control policy that is applied to OMP updates that are sent from a SD-WAN Controller to a WAN edge device and applied after the SD-WAN Controller uses the best-path selection algorithm.

Overlay Management Protocol (OMP) The routing protocol of the SD-WAN fabric. OMP is used to distribute all routing information, encryption keys, and other policy information. OMP runs inside a DTLS/TLS tunnel between a SD-WAN Controller and WAN edge devices.

P–R

packet duplication The process of forwarding a redundant copy of a traffic flow down a duplicate path in order to protect against packet loss.

preferred color An optional configuration argument that allows for the specification of a selected color or colors to use when forwarding a class of traffic, as long as those classes are compliant with the SLA class.

preferred color groups A ranking system for preferred and backup colors in application aware routing. You can configure up to three levels based on priority or path preference.

quality of service (QoS) The categorization and prioritization of traffic in a network, typically based on application type and requirements.

REST API An HTTP-based API that allows you to interact with SD-WAN

components via common HTTP commands. Code can be written to interface with this API.

role-based access control (RBAC) A policy-neutral access control mechanism that is defined around roles and privileges. The components of RBAC, such as role-permissions, user-role, and role-role relationships, make it simple to perform user assignments.

S

SD-WAN Controller The control plane of the SD-WAN fabric, which acts as a Border Gateway Protocol (BGP) route reflector and is responsible for distributing encryption keys as well.

service insertion The process of redirecting a network flow to an additional device for the purpose of performing a function on the traffic. Common network services include firewalls, load balancers, and caching engines.

service-level agreement (SLA) A commitment made by a service or application provider to customers for a minimum level of service or uptime.

service route A route that advertises a service, such as a firewall or an intrusion prevention system, to the rest of the network. Policy can be deployed to force traffic through these services.

SLA class list A list that allows an administrator to specify the maximum

Technet24

loss, latency, and/or jitter on an SD-WAN tunnel that a specific class of traffic is forwarded across.

SLA dampening A mechanism used to slowly reintroduce the transport back into the available paths for forwarding to ensure that you don’t reintroduce a transport while it is still having connectivity problems.

software as a service (SaaS) Software applications that are outsourced to providers and that typically run in the cloud.

software-defined networking (SDN) A network management approach in which network flows, rules, and operations are defined and deployed from a centralized controller rather than on each individual network device.

strict An optional configuration argument that specifies that the class of traffic should be dropped rather than forwarded in the event that no classes meet the required SLA.

T

Terraform An IaaS-based tool developed by Hashi Corp. that can be used to develop plans in a human-readable language called HCL. Terraform can be used to apply this configuration and also track the state of the configuration and alert to any drifts.

TLOC A Transport Locator route that distributes next-hop information and

also connects the fabric to the physical underlay. Data plane deployment can be influenced by manipulating TLOC information.

tloc-list A list element that can contain references to one or more TLOC routes described by their system IP address, color, and encapsulation. It may also include the optional arguments Weight and Preference.

topology group A centralized control policy that allows an administrator to configure hub-and-spoke or mesh policies with predefined workflows and fully customized topologies through the manipulation of route and TLOC route information at an SD-WAN Controller.

transit VPC A virtual private cloud (VPC) that can be used for connecting multiple geographically dispersed VPCs and remote networks in order to create a global network transit center. A transit VPC simplifies network management and minimizes the number of connections required to connect multiple VPCs and remote networks.

V–Z

virtualization A process in which applications and software are abstracted from the underlying physical hardware resources and run as virtual instances.

zone A group of one or more VPNs.

Technet24

Appendix A. Answers to Chapter Review Questions [This content is currently in development.]

This content is currently in development.