SD-WAN: A Technical Tutorial

A Technology Primer from CIMI Corporation covering the architecture and future of SD-WAN and Network-as-a-Service (NaaS)

This document is a tutorial on SD-WAN technology and its evolution, written by Tom Nolle, president of CIMI Corporation. Copyright © 2018, CIMI Corporation, Voorhees NJ USA. All Rights Reserved. Reproduction, distribution, and use of this document must conform to this license of use statement below, with any exceptions or deviations explicitly authorized in writing or email by CIMI Corporation. ***License of use****. This document can be read, posted, and distributed freely in its current PDF form, providing it is not edited in any way, providing that a link to it is accompanied by a statement attributing it to CIMI Corporation, and providing that the statement include a link to CIMI Corporation's website at http://www.cimicorp.com/. In addition, the reference, distribution, posting,* ***cannot in any way imply any commercial endorsement*** *by CIMI Corporation or Tom Nolle of a product, service, or vendor. The material herein expresses views on the SD-WAN space that are the personal and professional opinions of Tom Nolle and CIMI Corporation as of the time of original publication.* *We thank Dwight Linn, CEO at FAE Telecom Inc., for agreeing to proofread this tutorial.* ***No one*** *has influenced this material.*

Contents

A Little Background Music 4
The "Basic Overlay" SD-WAN 6
SD-WAN Addressing 9
SD-WAN Topology, Nodes, and Models 11
Extending the SD-WAN to the Cloud 13
What's "Logical Networking?" 15
Securing SD-WAN 17
SD-WAN Features for Virtualization 19
SD-WAN Management 21
Integration of SD-WAN: Architectures and APIs 23
A Summary of Important SD-WAN Features 26
The Future of SD-WAN 28
Using this Tutorial to Assess Vendors or Services 30
Some Final Thoughts 32

Software-Defined WAN (SD-WAN) is both the synthesis of and an evolution of a number of different technologies. It combines some of the features of "tunneling" and "overlay networking" that have evolved in networks for over thirty years, with the recent idea of defining network services largely through software rather than by assembling specialized devices. To most people, it's an evolution of software-defined networking (SDN), and in fact most of the SD-WAN products today are based on principles established by early SDN implementations.

All networks serve the basic mission of providing connectivity, the free exchange of information among the network service access points (NSAPs) that connect its users. Connectivity is based on the concept of **addressing**, whereby NSAPs are assigned a unique address, and data that's tagged with that address will be routed by the network to the NSAP it identifies. Decades ago, it was realized that both Level 2 and Level 3 networks could be used to create point-to-point "tunnels" or "virtual wires" that could then be used as the physical layer of yet another network. This second, higher-level, network is usually called an "overlay" because it sits above a traditional network and uses that underlay network only for transport connectivity. The real, or "logical" connections are created by the overlay.

In many of the early applications of overlay networking, including those SDN applications, the overlay provided a way of creating an **independent data center network** for applications or tenants. This overlay network's destinations were not directly visible to the underlay network users; additional routing of the overlay network address was required. SDN vendors focused on this data-center mission, and that's where most of SDN is deployed today.

The application of overlay technology in the WAN has emerged primarily to address a problem with enterprises' own private networks, usually called "virtual private networks" or VPNs because they are built not on private trunks and routers but rather on a carrier service, usually IP and Multi-Protocol Label Switching or MPLS. MPLS VPNs are usually associated with expensive Ethernet access services, and MPLS VPNs are expensive to deploy and maintain in terms of equipment and operations. That combines to make them unsuitable for both small sites and for sites in areas of the world where highly developed business network services are not available.

The Internet is available anywhere communications is available, and so if an overlay network could be created on top of the Internet, it could provide VPN services everywhere the Internet reaches. It would lack the kind of quality of service that MPLS VPNs provide to major sites, though, and so the initial model for SD-WAN was one that overlaid the SD-WAN's VPN service on top of not one network but two; MPLS VPN and the Internet. This service model was the first phase of SD-WAN, and it's the place to start our tutorial, but SD-WAN is much more, and we believe it will become the framework for full network virtualization, the "Network as a Service" everyone is waiting for.

SD-WAN is important for what it can do to extending VPNs everywhere, its "basic" mission as we describe it here. In that mission, it's an incredibly simple technology, and all SD-WAN vendors support the basic mission. SD-WAN's future, perhaps most vital, mission is a lot more complicated. It could define network-as-a-service and a big part of the future of networking. That future mission is where SD-WAN vendors separate. If your SD-WAN needs face the future that we at CIMI Corporation see it, it's that **future mission** that should define your SD-WAN procurements. We talk about both the present and future missions here in this tutorial, but you'll need to check with prospective vendors to see which vision they currently, or are committed to, support. Your SD-WAN decision could shape your networking future; get it right.

A Little Background Music

This tutorial is on SD-WAN technology, not on the general topic of networking. To make it useful to most of those who really need it, we have to assume the reader has a basic understanding of IP networking and at least a nodding acquaintance with the notion of software-defined WANs. Even some who do, though, might wonder how the whole SD-WAN thing got started, and so for those who are interested in a bit of philosophical history, we offer this tutorial section. Those who want to dive right into SD-WAN should feel free to skip it!

Broadband. That's what everything we see in networking is about, and what it all started with. Twenty-five years ago, the branch offices of businesses had perhaps a 56 **kilobit** per second of data bandwidth available to them. A few headquarters sites might have 45 Mbps, but they could pay as much as ten thousand dollars per month for the connection. Most had a 1.5 Mbps connection, perhaps two. Today, that 1.5 Mbps wouldn't even be considered "broadband", and consumers can get the equivalent of two 45 Mbps connections for less than a hundred dollars a month.

Voice-band let us call; broadband lets us **experience**, and everyone quickly realized that the Internet and broadband could deliver thousands, millions, of different experiences. Data connections of old not only were expensive and slow, they were **fixed**. Today, broadband access is truly a gateway, and the Internet and business networks (virtual private networks, or VPNs, based on Internet technology and the Internet Protocol or IP) have to be able to provide uniform connectivity not from point to point, but to and from every point.

IP exploded when the Internet did, supplanting older technologies like frame relay and asynchronous transfer mode (ATM), but IP really came about as a way to support bursty, variable, communications between large sites. Most Internet experts would admit that if the problem of today's Internet were posed without an IP in place to solve it, the technologists would have come up with something else. That's not an option today, with billions of devices in use that "speak" IP.

The edge of the network is fixed by the limitations of the things that use it. What's inside could be changed, though. As bandwidth continues to grow, and bandwidth demand does likewise, technologies that optimized capital investment in network equipment (network functions virtualization or NFV) and tried to do internal traffic management and bandwidth optimization differently (software-defined networking or SDN) came along.

SDN and NFV and most of the advances we've seen in networking were aimed at reducing **cost per bit**, meaning the cost of bandwidth. This focus is what's allowed us to reduce the price of multi-megabit networking so much that consumers today can buy what a major corporation could literally not have obtained at any price just 25 years ago. The quest for cheap bits first focused on optical networking, including "agile optics" to make connectivity more flexible, then on making the interior of IP networks cheaper, using those good-old SDN and NFV acronyms, along with white boxes and forwarding languages and all the rest. We've made incredible progress in lowering the price of bits, and there will be more to come.

The problem is that cheap bandwidth and "experience networking" mean you have to be able to deliver a lot of different experiences and do so cheaply. Servers and hosting ended up being another bottleneck to getting to the optimum future network, and so the information technology gurus came up with the idea of creating pools of resources to be allocated dynamically to applications and services when needed. This improves the efficiency of server hosting, the efficiency of experience creation. It was called "virtualization".

With virtualization, an application isn't assigned to a server but to a virtual or abstract entity. Developers can build their apps as though that entity was real, and operations people can talk about deployment of the entity the same way. Underneath, though, that entity is abstract until that deployment actually happens, which means that where it's put is the final step in the whole process.

That collides with one of the most fundamental truths of IP networking, which is that an IP address defines a specific place in the network, a geographical point. It doesn't define the identity or nature of what you put there. If something abstract like a virtual game server gets deployed, it gets an address based on the "where" not the "what". If it breaks and has to be moved, the address has to change. There are ways to do that, but they have pitfalls, not the least of which is a potential significant interruption of service.

The net of all of this is that virtualization, things like virtual machines (VMs), containers (Docker, for example) and the cloud (OpenStack, Amazon, Google, Microsoft, and more) have all complicated our old connection problems by making stuff that used to be "real" like a server into an abstraction that can be mapped to any convenient host, anywhere. What good is addressing things to fixed locations in a world where nothing is really fixed, or ever likely to be again? We need virtual networking to accommodate virtual resources.

Virtualization in networking isn't new, but it isn't complete either. We already know that the first step in creating an agile, user-and-experience-responsive, virtual network is to make it software-based, software-defined. SDN does that inside the network, and in the WAN the solution is the Software-Defined WAN or SD-WAN.

SD-WAN is an accidental solution to the fundamental problem of making networks virtual, which is the resolution of that old "where it is versus what it is" problem. It started off as a way of connecting remote offices cheaply, but the technical features needed to do that also created the first wide-area networking advance that separated addressing from network location. Build an abstract layer of services on top of IP, with its own names and rules and policies, and map it to whatever network resources we have.

SD-WAN may have started out as a way to connect remote locations cheaply, but it's going to be a lot more, and that's why you need to understand it. That's why CIMI Corporation is offering this tutorial, as an open document that can be distributed freely under the license agreement on the front page. Even today, SD-WAN discussions focus too much on that very limited first mission of branch connectivity. We think a lack of understanding of the technology's real capabilities is the reason, and so we're dedicating this document to building that understanding and helping SD-WAN reach a goal it's uniquely qualified to support.

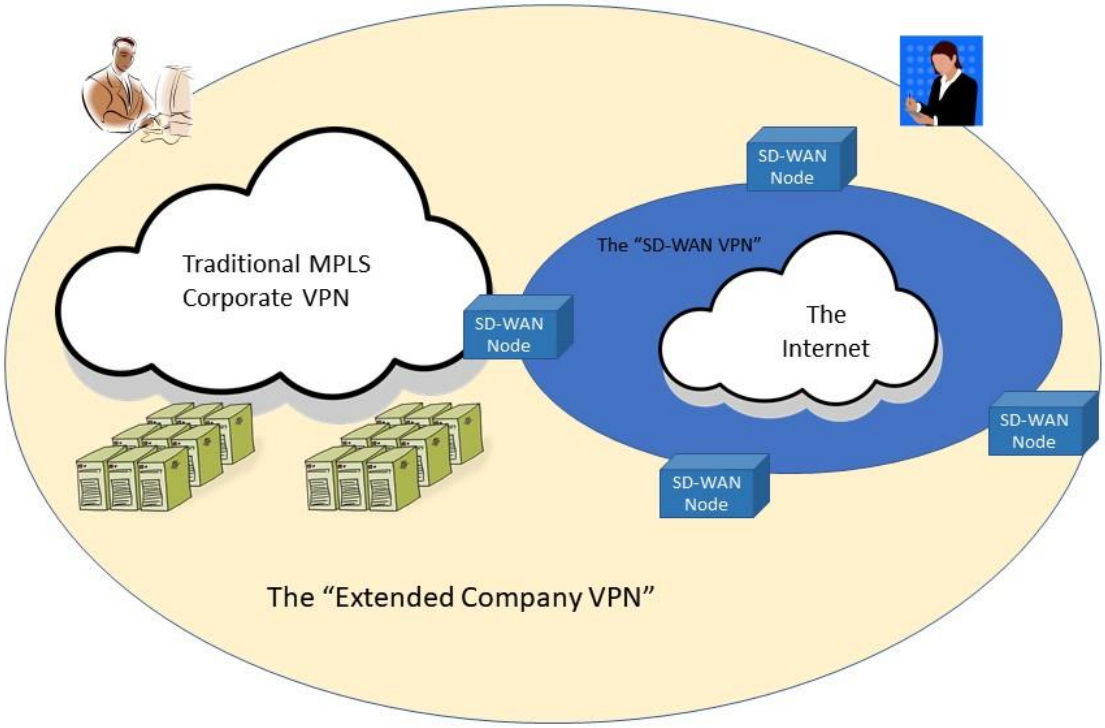


Figure 1: The Basic SD-WAN Model

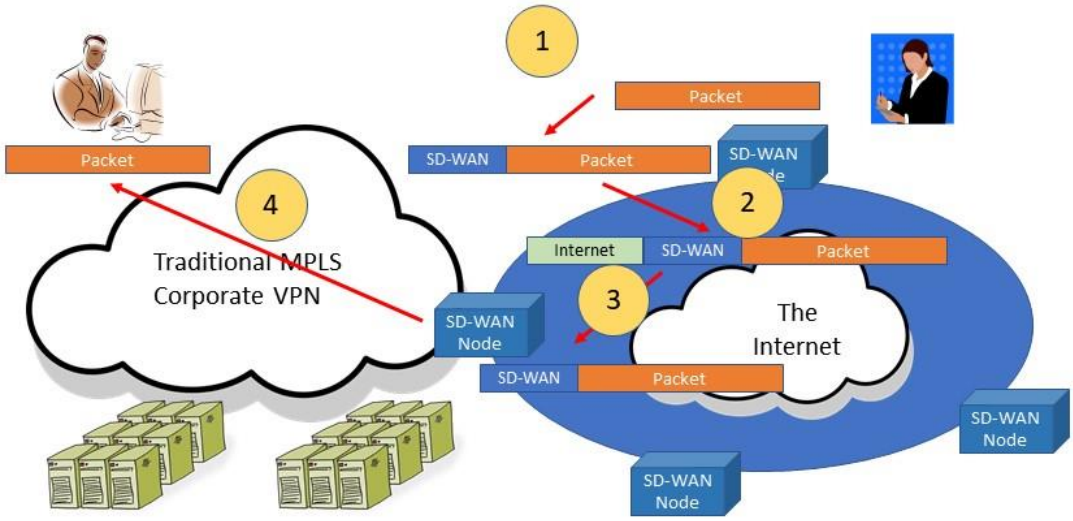
The "Basic Overlay" SD-WAN

SD-WAN today is deployed in a very specific way, so let's start our discussion by explaining that current, basic, approach, as shown in Figure 1. The figure shows, on the left side, the traditional MPLS VPN. Since it's expensive or impossible to use MPLS for all remote sites, SD-WAN creates a second network that we'll call the "SD-WAN VPN" (the right side of the figure). This VPN is based on the same IP address used by the company's MPLS VPN, but the users aren't directly connected to MPLS. Instead, they are connected via the Internet to an "on-ramp" or hub that links the SD-WAN VPN and the "normal" MPLS VPN. The SD-WAN VPN and the MPLS VPN combine to create the "Extended Company VPN".

Inside any SD-WAN VPN is one or more transport networks that provide the actual connectivity. In nearly all cases in today's market, the Internet is the transport network. The SD-WAN VPN is created with a series of "SD-WAN nodes" that connect with each other via the Internet and provide SD-WAN VPN users with connectivity. Within this community, inside these nodes, SD-WAN VPN routing and networking features apply. We'll keep making that point because it's critical in understanding SD-WAN.

The principle of routing in an SD-WAN VPN is almost identical to the way IP routing works overall. Think of each SD-WAN node as creating a "subnet" on its user or port side (some vendors may support

distributed subnets), representing the collection of users (human or otherwise) that are in the SD-WAN address space and connected to that node. The node itself also has an address on one or more underlay network services. These addresses are only used by SD-WAN nodes to pass traffic to other SD-WAN nodes, not directly by SD-WAN VPN users.



1. A user sends data to the network (the SD-WAN VPN)
2. The SD-WAN Node adds the SD-WAN header, which identifies the real destination user on the MPLS VPN
3. The trunk process of the SD-WAN node adds the Internet header addressing the SD-WAN hub to the MPLS VPN
4. The SD-WAN hub node strips the Internet and SD-WAN headers and sends the packet to the MPLS VPN user address

Figure 2: How the SD-WAN Overlay Works

Look now at Figure 2. When a data packet is sent by a user on the SD-WAN VPN, it first goes to the SD-WAN node it's connected with. There are three possibilities at this point. First, the address might be on the Internet, not part of the VPN at all. This traffic would be passed through without further processing. Second, the address might be on the MPLS VPN. In that case, the traffic would be routed to the **hub** that connects the SD-WAN to the MPLS VPN.

The third option is that the address is on the SD-WAN VPN. In that case, the node will look up the SD-WAN destination address in a table, and the table will provide the identity of the underlay network used to reach that destination and the address of the destination's SD-WAN node. The SD-WAN VPN source and destination addresses and other information will be added as a **header**, along with the header and addresses for the underlay network, and the packet will be sent through the correct interface, to the address provided.

In the basic SD-WAN model where an SD-WAN **extends** the MPLS VPN, most packets sent by SD-WAN VPN users will probably go to destinations on the MPLS VPN. In this case, the packet is addressed to the MPLS VPN address of the user, which SD-WAN routing knows is accessible through an SD-WAN **hub**

node. The Internet address of that hub node is added and the packet is dispatched over the Internet. When it reaches the addressed hub, the Internet address is stripped off along with any other SD-WAN header data, and the packet is sent through the MPLS VPN to its destination.

If the destination were on the SD-WAN VPN rather than on MPLS, the process is the same, except that the destination SD-WAN node would be the node to which the destination connects, not a hub. Thus, within the SD-WAN VPN, the Internet connects nodes and the SD-WAN VPN connects users/resources.

Different SD-WAN implementations use SD-WAN nodes differently, and the biggest factor in how SD-WAN nodes are used is whether the model of deployment uses SD-WAN only to **extend** an existing MPLS VPN or to create an **overlay** that includes MPLS VPN transport. We'll explain that in the SD-WAN addressing SD-WAN Topology sections below.

The goal of the basic SD-WAN implementation is to extend the company VPN, most often by using the Internet. It's also possible to replace the MPLS VPN completely and use SD-WAN to connect everything. Other options include using multiple ISPs for diverse routing, overlaying multiple MPLS VPNs where several different providers have to be bridged to create unified connectivity, or all of these combined. These different configurations really only extend the principles of the basic model, and this approach represents one thread of SD-WAN feature evolution we see in the market today. What are the other threads, and how does the SD-WAN industry follow them?

SD-WAN Addressing

The basic model isn't the only SD-WAN model, even in today's market. To understand the other approaches to SD-WAN, we need to start with the general notion of **connectivity**. Everything in networking is built on top of "connectivity", and connectivity means addressable users and resources, which in turn means addresses and address spaces. A "network" always has an address space, a set of addresses it's prepared to route among destinations. The Internet is the foundation of IP addressing, and it defines both "public" and "private" IP addresses. The former are assigned uniquely from the same address space (IPv4 or IPv6), which means that companies can receive a range of addresses, an address space, from the public set and use it for their own networks. Private IP addresses are assigned from a range of reserved addresses (defined for IPv4 in RFC 1918).

When a private IP address is used, as it commonly is within a home network for example, a gateway device adds a new header that contains the private address to a data packet, then hands the packet to the network using the IP address of the gateway. The destination routes the response back to the IP address it sees, which is the gateway address, and the gateway then strips off that address and uses the private address to reach the user.

IP networks are actually networks of networks, starting with a "local subnetwork" or "subnet". Most IP networks are built up from a community of users or resources connected on a LAN. This LAN lets all the members of the community "see" each other, and there are simple connection rules that apply within the community. Every home network is an IP subnet, many multi-component applications deploy into data center subnets, and nearly all cloud applications run in subnets. To get out of a subnet, to reach other users or resources, traffic passes through a **gateway**, which is a boundary between the simple subnetwork connection model to the broader IP model used in the Internet or in an MPLS VPN.

This same principle is used for SD-WAN, whether public or private addresses are used. SD-WAN subnets are served by SD-WAN **nodes**, which act as those gateways in providing access to things outside the local subnet. We'll say this many times in this tutorial because it's important: **where an SD-WAN node goes, so goes the SD-WAN VPN**.

The SD-WAN VPN address and routing work only within the SD-WAN VPN. Within the SD-WAN VPN, traffic is routed based on an SD-WAN header, and SD-WAN nodes are connected over the Internet, using the IP address of the nodes not the users. As Figures 1 and 2 show, this means that while the SD-WAN VPN and the corporate VPN are connected by a hub node to create a "company VPN", SD-WAN routing controls traffic in the former VPN, and it's possible to add features to the SD-WAN VPN without changing how the MPLS VPN works. This allows SD-WAN connectivity features to evolve beyond those of normal IP networks, within the SD-WAN VPN. Outside that VPN, the old rules still apply.

Whether public or private IP addresses are used, an SD-WAN VPN can use the corporate (MPLS) VPN address space (which itself can be public or private) or it can create a totally new address space. Most SD-WAN implementations today will use the corporate VPN address space and will be designed to **extend** the VPN to sites that, for reasons of economy or availability, are not connected via MPLS.

In this model (the basic model shown in Figure 1), MPLS VPN users communicate with each other as they always have, and SD-WAN VPN users connect to the corporate MPLS VPN via one or more "hub"

SD-WAN nodes. These nodes manage the SD-WAN overlay header to ensure that data moving within the SD-WAN is properly addressed, and that SD-WAN header data is removed when data exits the SD-WAN VPN for an MPLS VPN destination.

A likely future model for SD-WAN VPNs is the **overlay** model, in which the SD-WAN VPN expands to connect all users and resources, effectively becoming the company VPN. The current MPLS VPN and the Internet then provide underlay or transport connectivity. If the SD-WAN deployment replaces MPLS VPNs completely, the overlay model is mandatory. This model is more disruptive to deploy because every user and resource has to connect to the SD-WAN VPN, but it also lets the SD-WAN manage all connectivity, independent of IP network rules. We'll talk more about the "extend" and "overlay" models in the next section.

IP networking is constrained by the massive number of installed devices and IP-specific interfaces on laptops, phones, and servers. SD-WAN is a new layer, sitting between those devices/users and existing IP networks. That layer can adopt its own rules and offer its own features, as long as they are harmonized with traditional IP in the "port" or user direction, and the "trunk" or transport network direction. As SD-WAN features evolve, and SD-WAN connectivity looks more and more like the long-awaited "network as a service" or NaaS, it is likely that the overlay SD-WAN model will become more popular, and perhaps even universal.

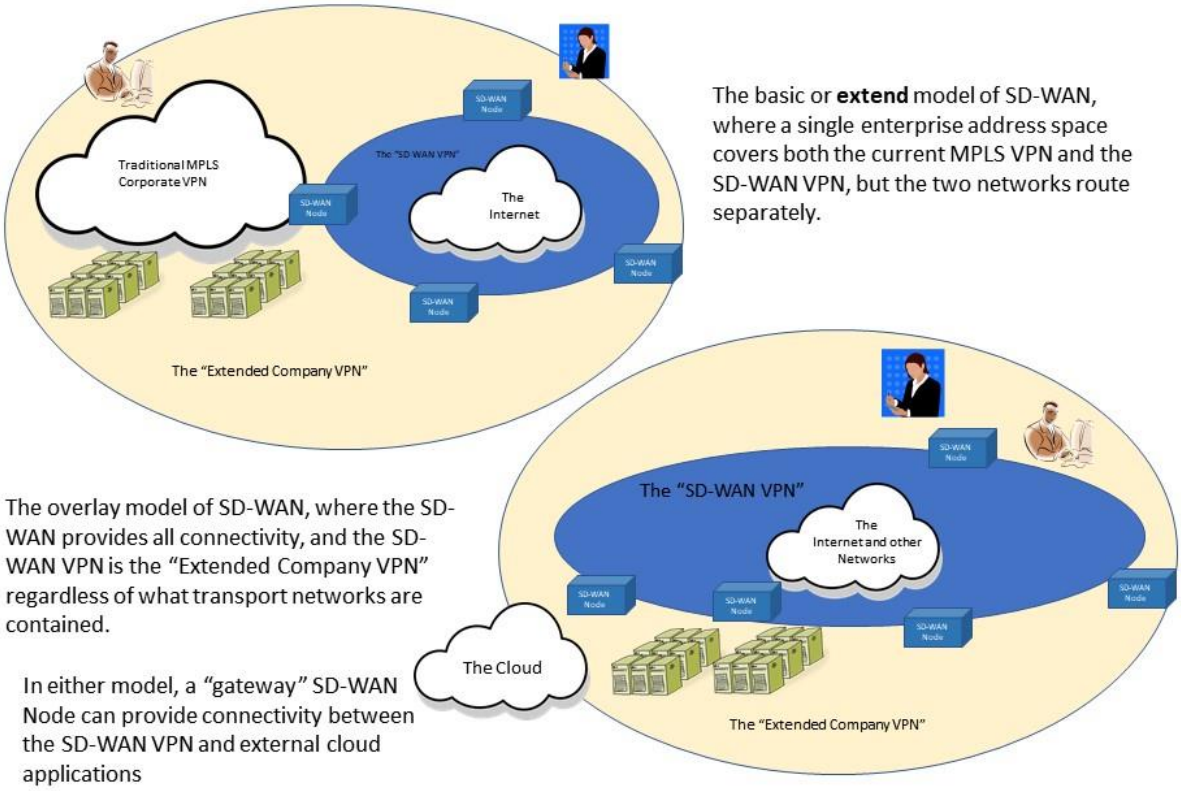


Figure 3: The Topology Options for SD-WAN

SD-WAN Topology, Nodes, and Models

As we've noted in prior sections, an SD-WAN is created by what we're calling (generally) "SD-WAN nodes". These are real or virtual network devices, and like all network devices they are deployed in a cooperative relationship. The way the nodes relate to each other, to the SD-WAN VPN, and to the underlay networks, depends on the specific implementation. We'll call the nodal structure and relationships the "topology" of the SD-WAN. Figure 3 outlines the high-level options.

The first of our deployment models is the "extend" model where the corporate VPN is the sum of the connectivity provided by the SD-WAN VPN and a corporate MPLS VPN. In the "overlay" model the SD-WAN VPN provides **all** the connectivity, so both Internet and MPLS VPNs (where used) provide only transport services.

There are three broad classes of possible SD-WAN nodes, and the way these classes are used (or if they're used at all) determines the topology options:

1. **Edge Nodes**. An SD-WAN edge node is deployed at a point where users or resources connect to the SD-WAN VPN, so they have a "port" direction of connection. Each edge node also has one or more "trunk" connections to the transport networks that are used to reach other locations. Every SD-WAN will have edge nodes, but users on the MPLS VPN may not have their own edge

nodes; they may enter directly. If an SD-WAN is used as an alternative to MPLS, then all users and resources will be connected to an edge node. That is also the case if the SD-WAN overlays **both** the Internet and the MPLS VPN.

2. **Hubs**. Some SD-WAN implementations will have SD-WAN nodes whose role is to connect the SD-WAN VPN across multiple underlay transport networks. Edge nodes could also serve this mission, but a hub would be designed explicitly for the function of cross-connecting the MPLS VPN with the SD-WAN VPN.

3. **Gateways**. A gateway could be viewed as a special class of hub node. It can serve two related roles. First, it can provide a way for an external IP network, like the Internet, to be accessed by the SD-WAN VPN. Second, it can provide a means for remote edge nodes to connect back to the SD-WAN VPN. This is helpful when the Internet is used as a means of connecting mobile users, or when public cloud services have to be connected and can't be made into edge nodes (see the later sections on the evolution of SD-WAN cloud connectivity.

In the "extend" model, the SD-WAN edge nodes are used to provide SD-WAN VPN connectivity where the MPLS VPN doesn't connect. The two VPNs share an address space and collectively create the enterprise VPN. In an overlay model, SD-WAN edge nodes are used everywhere to provide connectivity. The Internet and any MPLS VPNs or other network services are **inside**, acting only as transit networks. The SD-WAN VPN completely controls connectivity. A gateway node could still be used to connect remote users/resources, such as cloud-connected users.

The important point here is that the SD-WAN VPN is created by SD-WAN nodes, which encapsulate traffic in an SD-WAN header on entry and remove the header at exit. Every connection made in or through the SD-WAN VPN must pass through a node that serves as the "on-ramp", where header/address data is added for transport routing and where SD-WAN features are provided. Each must also then pass through another node, an "off-ramp" where the headers are removed and normal IP routing behavior will then apply. Where SD-WAN nodes go, so goes the SD-WAN VPN, and that's important to remember when you consider applications in the cloud.

Vendors may or may not call their nodes what we call them here, and in fact some may offer all of the features we describe within a single node type. The point is that an SD-WAN VPN has a number of different "node roles", and vendors will provide nodes (in device or software form) that fulfill those roles. If you can put an SD-WAN node somewhere, the SD-WAN VPN goes with it.

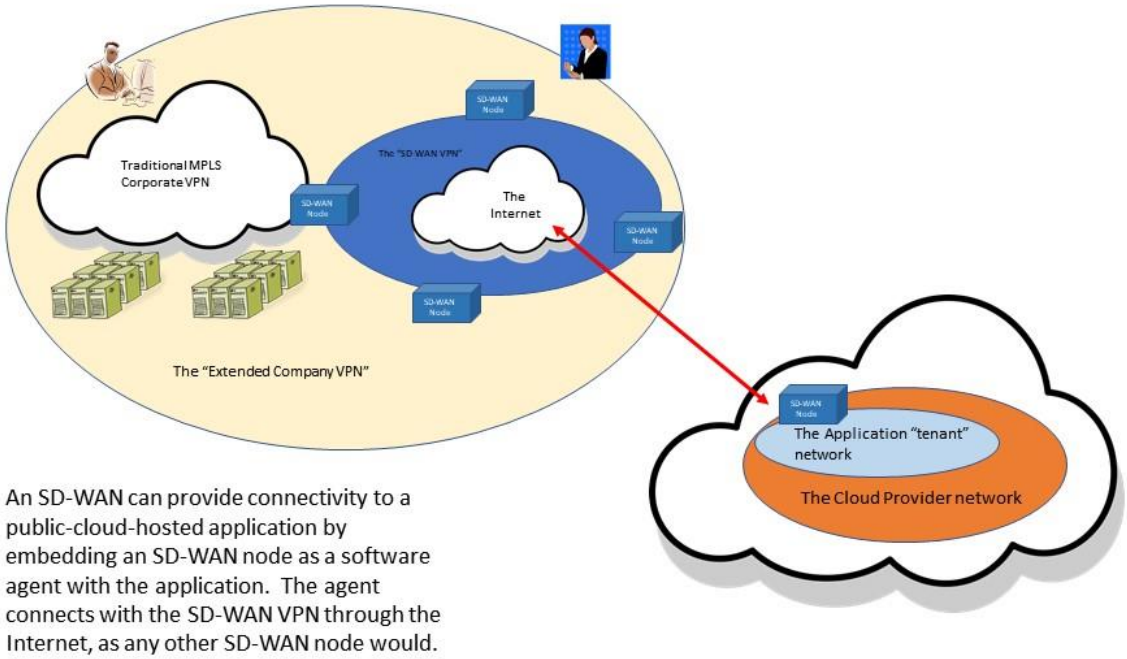


Figure 4: Cloud Connectivity with SD-WAN

Extending the SD-WAN to the Cloud

The basic model just described is fine for basic VPN connectivity among business sites, but it has a problem when some of the connections you need are inside a public cloud provider. It's expensive and difficult—sometimes impossible—to link a corporate MPLS VPN to a cloud provider. Generally, you can't put SD-WAN boxes inside someone else's data centers either, so what's needed is something that can provide SD-WAN node functionality in software form—an SD-WAN Agent that acts as an SD-WAN node to add a cloud process to an SD-WAN VPN. Figure 4 describes the basic approach.

The basic approach to extending SD-WAN to the cloud is simple. Remember from prior sections that an SD-WAN VPN is a network of SD-WAN nodes, which means that anywhere you can put an SD-WAN node that's connected to the Internet (or whatever the SD-WAN VPN transport network options are) can join the SD-WAN VPN. Put a node in the cloud, which means hosting an SD-WAN node agent component there, and you've got cloud support. How that works exactly will depend on the way the cloud components are addressed.

Public cloud services will assign IP addresses to the components/applications they host. These can be from a private IP address pool, a public set of IP addresses assigned by the cloud provider, or perhaps even from a public address pool owned by the customer; it depends on the cloud provider. Changes in

deployment can result in changes in the IP address, which may be handled through a cloud-provider DNS or "elastic" IP addresses that follow redeployments.

The goal of basic SD-WAN cloud support is to create within the public cloud, an SD-WAN subnet controlled by an SD-WAN node hosted in the cloud. The resulting subnet is joined with the SD-WAN VPN and in all respects becomes a member of the enterprise VPN that overlays the SD-WAN. SD-WAN routing, logical addressing, policies, and so forth then apply to the cloud subnet, as they would to a branch office.

The exact capabilities of this subnet vary significantly by implementation, but the minimal feature set establishes a collection of cloud-local elements under the control of an SD-WAN node. This makes the components hosted within that collection appear as members of the SD-WAN VPN. There can be an SD-WAN "cloud subnet" for as many clouds as needed, including premises clouds, but the basic features for cloud support limit the value of SD-WAN in complex configurations.

As the figure illustrates, the easiest way to think about SD-WAN cloud support is to consider a collection of cloud applications or components as a "subnet", not unlike the subnet you'd find in a typical branch office. When SD-WAN cloud support deploys, an SD-WAN node is added to this cloud subnet, and like all SD-WAN edge nodes, it then provides connectivity between the subnet users (which in this case are cloud application components) and the rest of the company VPN.

The biggest problem that basic SD-WAN cloud support faces is the elasticity of cloud resources, a benefit cloud users rely on. "Elasticity" means failover across hybrid and multi-cloud configurations, as well as scaling in the same dimensions. All this implies considerable direct virtualization support, something most SD-WAN vendors do not offer.

Another problem is the variability of cloud services. The old days of nothing-but-IaaS are gone; we now have platforms, containers, functions/microservices/lambdas, web services, and more. Complex cloud services have complex addressing requirements, and since SD-WAN vendor implementations vary, it's important to check to see exactly what form of cloud support a given SD-WAN solution provides.

You are probably not surprised to hear that while cloud extensions to SD-WAN are the most common of the features beyond the basic model of Figure 1, they are also the most variable in terms of how they're implemented. It is very important to understand what your prospective SD-WAN provides for cloud connectivity, and what specific public clouds they can provide reference architectures or use cases for.

The cloud, of course, leads to the broader (and even more complicated) question of supporting virtualized infrastructure. We'll get to the way SD-WAN could fully support the virtual world in a later section.

Figure 5: A Sample Logical Name Hierarchy

What's "Logical Networking?"

The short answer is "everything" if you believe (as we at CIMI Corporation do) in the future of SD-WAN as the best hope for network-as-a-service (NaaS). It's also the key to expanding SD-WAN beyond the basic branch and basic cloud support we've already discussed. Logical networking isn't about IP or Ethernet or any of the stuff we see in physical networks. It's about users and applications, resources and resource consumers. It tries to get all the technical stuff, the things important in networking today, out of the way. Isn't that what networking is supposed to be like?

The idea behind logical networking is that since SD-WAN creates a VPN and its own rules for routing, why not give it true what-it-is-where-it-is independence, something that IP networks have been shooting for almost from the first. Today, addresses get packets to a location. In the future, with logical networking, packets can be routed to an **identity**, a specific user or resource. SD-WAN figures out the details.

Addresses are inconvenient to use, which is why we use URLs, which are something like real names. Logical networking can take URLs further, **making** them real names with all the properties of naming resources and users that we'd find convenient. Foremost among those useful properties is the notion of **hierarchical naming**. Instead of having some arbitrarily assigned IP address as the way a user or

resource is known, why not “Bill of Accounting” or “Sally in New Orleans”. For resources, something like “CRM” or “Payroll in Personnel” would be nice. Obviously, it would be helpful to have multiple layers of hierarchy, so you could have “Sally in Accounting in New Orleans”. See Figure 5 for details.

There are a lot of ways of making hierarchical naming work, more than we could discuss here without getting into specific SD-WAN vendors and implementations. Instead of talking about the “how”, we’ll look instead at the features, beyond the essential one of hierarchical naming, that might be available now or in the future.

One we’ve hinted at is the notion of “roles” for users and “classes” for resources. A worker might be assigned one or more roles, and a resource (an application or database that’s accessible online) might be associated with one or more classes. If policies can relate roles to classes, they can define how a group of users can access a set of resources. Is the exchange allowed or barred, high priority or low, encrypted or plain-text? Roles and classes, either as part of the hierarchical name or as a feature of a name or even part of a name, can form the foundation of security and compliance for SD-WAN.

Another thing SD-WANs could or do offer is **registration** in some form. Registration means assigning a logical name to the entity it belongs with. You could make the assignment based on something static, like the MAC address of a device, you could parameterize it in as part of application deployment, you could have an authentication dialog provide it…you get the picture. The more flexible the registration options are, the more valuable hierarchical naming is, and the closer you come to true logical networking.

Continuing on our advance into more improved features, consider that a logical name could let a user’s privileges follow them as they move from their office/cubicle computer system to a phone/tablet, to a home computer. More significant, an application’s logical name could follow it as it gets redeployed during a failure. For scaling, logical names could be associated with whether an application/component was scalable under load, and even provide or integrate with load balancing among instances.

If we combine these features with **session awareness**, meaning the ability of the SD-WAN to recognize a connection relationship between parties (users and applications, for example), we can go even further. Obviously sessions could be policy-managed, permitted or blocked by policy depending on the nature of the application and the identity or role of the user. Obviously sessions could be assigned a priority, and they could even be routed differently depending on priority. That routing difference might involve selecting a different ISP, a different hub “on-ramp” to the MPLS VPN, etc. Finally, if a session is known to the SD-WAN VPN, its health can also be known. That means that a session could be rerouted if the primary path fails. Obviously this works best if there are diverse connection options available rather than just routing options within the same ISP.

All these features create a connection service that is based on identity not location, that uses available transport resources optimally, that can support redeployment and scaling. That’s the transformation that gets SD-WAN logical networking to NaaS.

Securing SD-WAN

Security is one of the many topics in networking where hype has obscured at least some of the basic truths. The goal of security is fairly simple; keep access to applications and data restricted to those who are entitled. In the real world, we've tended to divide this goal into two pieces—the protection against outside hackers (security) and the restriction of confidential information even inside the company (compliance, governance).

Outsider protection, the "real" security, is also multifaceted. There's intercept security, designed to protect information in transit on the network, access security that limits the ability to connect with applications and resources, and protection against denial-of-service attacks created by floods of traffic directed at a given NSAP.

One of the things that makes security such a problem today is the fact that an IP address isn't a definitive indicator of what/who something is. Most users are assigned IP addresses from a pool, and they change on a regular basis. In addition, the Internet doesn't enforce the accuracy of a "source address", so you can't really be sure whether even the IP address on a given packet is authentic. The nice thing about SD-WAN is that it can enforce rules of its own, including the reliable association of IP addresses with users and resources via logical naming. That facilitates writing connection policies that add considerably to the security, at least within the SD-WAN VPN.

Everything isn't on the SD-WAN VPN, of course. Outside users on the Internet aren't on it, and neither are users who, in the basic model of using SD-WAN to extend the company VPN, are already connected by MPLS VPNs. Traffic that doesn't go through SD-WAN nodes cannot inherit SD-WAN security features, and some external and traditional security tools may still be required.

Traditional security mechanisms like firewalls and encryption can still work within the SD-WAN, as long as the SD-WAN header itself is not impacted. In most cases, firewall tools will not pose a problem, nor will end-to-end HTTPs-type security, which encrypts the data payload. Check with each prospective SD-WAN vendor to be sure that current security tools will work, and to find the best place to connect them.

The most important thing to remember about SD-WAN security is that the SD-WAN VPN can itself be a big step forward in security in its broadest sense, meaning both outside and worker (compliance) access. Remember that the whole purpose of those early overlay SDN strategies was to provide tenant isolation in cloud data centers. A good SD-WAN creates a single-tenant network (with you the buyer as the tenant), and the resources and users on the SD-WAN VPN are not directly visible on any of the underlay networks. SD-WAN routing could also, in theory, limit the Internet access of internal users, or limit the ability of Internet users to access company resources. Thus, it's wise to understand exactly what security features an SD-WAN can provide and integrate them into your security planning.

The same is true for compliance/governance of internal access to resources. SD-WAN routing could offer some access policy control for all resources, providing that the SD-WAN maintains complete routing tables for internal resource destinations. A user, or a group of users who have common application access needs, can be connected to what they need while the rest of the company simply has no route to get to them.

Remember that doesn't mean that SD-WAN directly secures you against everything. SD-WAN VPNs are VPNs, and in most VPNs there's a gateway mechanism to allow for Internet access. Intruders can still enter through that gateway and access any resources that are "public" meaning that they are intended to be exposed to online use. Today, many companies keep their web servers outside their VPNs, connecting the two through a "demilitarized zone" or DMZ to keep the company's public world and private world separate. You may still need to do that.

Also remember that in many SD-WAN configurations (most as deployed today), SD-WAN VPNs **extend** the corporate MPLS VPN, and so MPLS VPN traffic may not go through any SD-WAN node. If there is no universal overlay, as there would be if everything connected via an SD-WAN node, then full control over security and compliance has to come from outside the SD-WAN. Thus, policy controls over connections would not be available to the "extend" applications of SD-WAN. A full overlay implementation could control access everywhere.

Mixing security mechanisms can lead to confusion, and this is particularly true when SD-WAN is extended into the cloud. In most cases, cloud-hosted resources will enter into the SD-WAN VPN via a cloud agent process or an SD-WAN gateway. This means they can be secured and have access control augmented by SD-WAN features. Because "normal" MPLS VPN sites may not be connected through an SD-WAN node, access control at the user point of connection won't be provided by SD-WAN. If you want SD-Wan access control and security for cloud-hosted resources, you'd need to have an implementation that let you control "incoming" connections from any source. Even then, you won't get the same of logical control as you'd have if everyone was on an SD-WAN overlay VPN directly.

The best strategy for SD-WAN security is to start by using every tool that the SD-WAN can offer to secure information, users, resources, etc. Where that's not enough, build out additional security layers, always being mindful of where you put them in the SD-WAN configuration. Traditional tools will work outside the SD-WAN VPN, but many won't work within it because of the additional layer of networking the SD-WAN VPN creates.

SD-WAN Features for Virtualization

The need to support access to public cloud resources (applications, components, and databases) can be provided at a basic level by SD-WAN VPNs as described in an earlier section (see Figure 1). That doesn't mean that everything needed to optimize the value of the cloud is a simple matter of adding a cloud-hostable SD-WAN edge node. The cloud is one of a widening range of virtualization technologies most users will need to accommodate eventually. Even just using the cloud optimally will require accommodations at the network level. SD-WAN VPNs can help with both.

Virtualization is a strategy that abstracts function from function implementation. From "above" (usually the user side), something "virtual" looks like a traditional resource or infrastructure element. It's really a model or abstraction that is mapped to (implemented on) a specific combination of resources, including devices, servers, software, etc.

If something breaks in the original implementation of an abstract function, the implementation can be redeployed, even in a different area or using different tools. If the implementation is a true representation of the abstract function, the user won't see a difference (other than perhaps a short outage during redeployment).

Even unexpected workloads can be accommodated with virtual resources. If function implementations are designed to be deployed in multiple copies to absorb work, then this scaling can let you add resources when they're needed and remove them when you don't.

The problem that virtualization creates is that everything in networks requires a connection, and if an abstract function is redeployed differently or scaled, then reaching the new piece of the functional puzzle will require connectivity. You really need fully virtualized networks to support fully virtualized infrastructure.

Redeployment or scaling doesn't change what something is, but it sure changes the "where" side. There are a number of ways that you can reconnect things after a redeployment or scaling to reflect the new locations of applications/components, but they all involve a number of steps and may require some form of automation (DevOps scripts, zero-touch automation) to complete often without errors. These problems come about, as we've already noted, because in traditional IP networks, an address identifies **where** something is, not **what** it is. Move the "something" and you change the address.

An SD-WAN VPN doesn't have to follow the rules of traditional networks. It's possible for an SD-WAN to provide true **logical networking**, where both users and resources have an "identity" that is used to map traffic to that user/resource to its current location. See our section on "What is Logical Networking" earlier in this tutorial. Logical networking not only addresses the challenges of virtualization and networking, it also addresses the growing challenge of mobile users/workers, and can even improve security and compliance.

It's difficult (or impossible) to apply logical networking to traffic that **doesn't** pass through an SD-WAN node, so check this feature carefully with vendors if you have a configuration where SD-WAN extends rather than overlays current MPLS VPNs.

There are a number of virtualization-related features that might be offered in an SD-WAN implementation; check with your supplier to find out exactly which of these are provided.

1. **Container networking support**. SD-WANs could treat container subnetworks the same way as cloud subnetworks, and this could enable an SD-WAN to control traffic inside multi-component container deployments and access to exposed (public) container addresses. If an SD-WAN vendor supports container networking, it's easiest to adopt if the vendor provides an SD-WAN agent integrated with the container orchestration tool (Kubernetes, for example).

2. **Specific failover support**. Failover requires that the replacement resource be addressed by users, which either involves DNS updating or the use of a logical, location-independent, address. Many public and private cloud, container, and virtualization resources include some failover support, so if you plan to use failover features of an SD-WAN, be sure they'll work with the virtualization tools you're employing.

3. **Scaling and load-balancing**. A few SD-WANs will provide for load-balancing to distribute work to multiple instances of a component, and manage the process of connecting new instances during scale-out and removing them during scale-in. Again, you'll need to understand how this feature works for your specific virtualization/deployment model.

4. **Hybrid and multi-cloud**. Hybrid cloud and multi-cloud configurations complicate virtualization support because the address management and orchestration capabilities are usually at least slightly different among cloud providers and on the premises. It's easier to control these issues in hybrid cloud configurations with one public cloud partner, but with multi-cloud it may be difficult to harmonize on a common approach. It may be necessary to shift from a cloud provider tool for orchestration and address management during fail-over or scaling, to deploying your own approach in the cloud. Consult with SD-WAN vendors on this; some already offer good multi-cloud integration.

If ever there was an area where close examination of SD-WAN vendors' capabilities is critical, virtualization support is that area. Very few enterprises will avoid a significant commitment to virtualization, hybrid cloud, and multi-cloud in the future. Likely none will avoid container use. That means that whatever SD-WAN strategy you adopt will need, at some point, to support all the virtualization options on the market.

A parallel truth is that if ever there was an area where vendor implementations of SD-WAN vary widely in capabilities, it's support for virtualization. Most current SD-WAN products offer little more than extension to the cloud, and very few provide any of the logical networking/naming features essential in resolving that old "what" and "where" problem. This is an area where you'll need to look very carefully at the features and plans of prospective SD-WAN sources. Otherwise your current SD-WAN choice could impact your ability to adopt what might well be the most important tool in application deployment and data center efficiency.

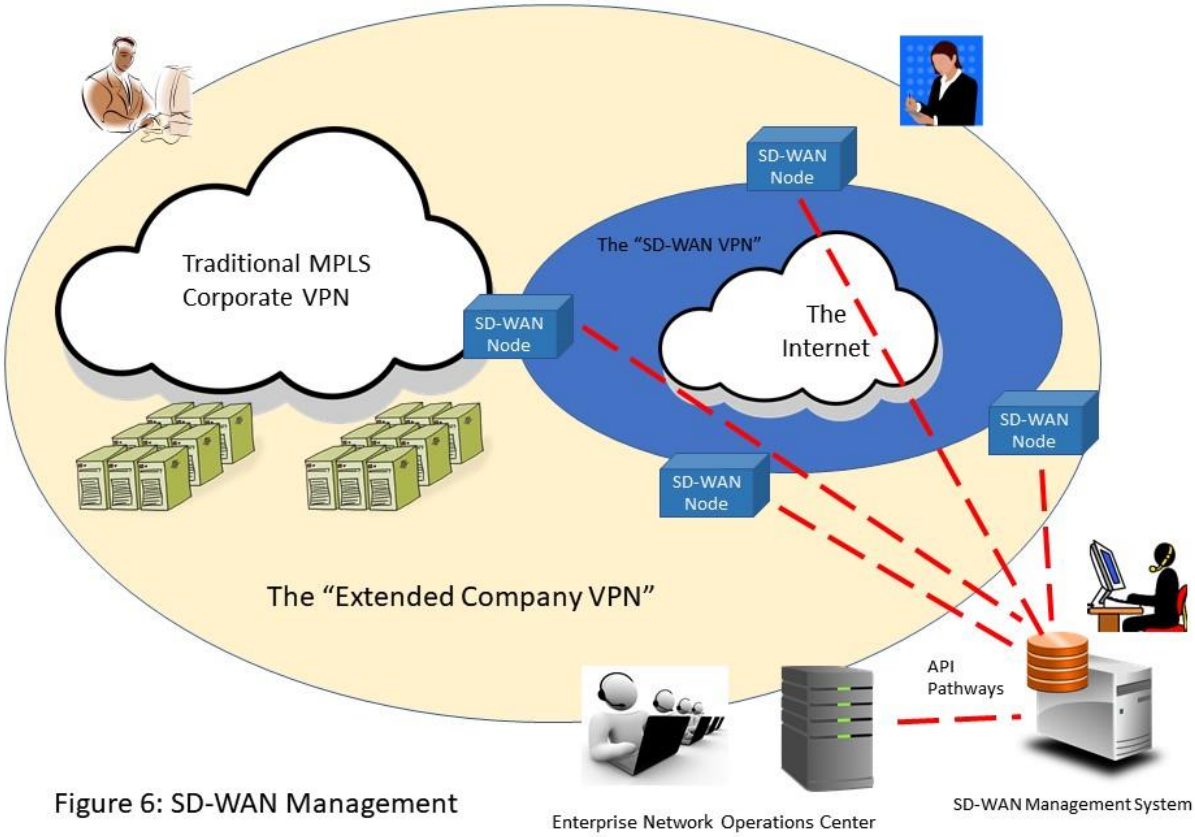


Figure 6: SD-WAN Management

SD-WAN Management

SD-WAN nodes are network elements, and every network element needs to be managed. Most will require parameterization, policy control statements, and all will have to be able to report its health in the form of operating status and parameters needed to measure behavior against a service-level agreement (SLA). Thus, you can expect that SD-WAN implementations will offer a per-node management connection to a management console, and the console is likely to expose management APIs for integration (see the next session). Some per-node management APIs may also be offered. See Figure 6.

Depending on how SD-WAN is sold, the management properties may either be exposed to the user directly or "subducted" into a service provider relationship. Products directly sold to enterprises are usually expected to be managed by the buyer, but service providers (including managed service providers or MSPs) and integrators or resellers may also provide management services to their buyers.

SD-WAN services sold by network operators (telcos, cable companies) and managed service providers (MSPs) are normally sold as managed services, meaning that the supplier takes on the responsibility for maintaining the operation of the SD-WAN VPN and meeting some specific service-level agreement (SLA) set by your service contract. Where this is the case, it's important to know just what management

features are still available to you as the user of the service, particularly if the SD-WAN product offers advanced features for logical names and policy management of connection relationships. Going through a third party for these features could be inconvenient, and in some cases even create compliance issues.

Whatever the source of management, you need a good management system. A good SD-WAN management system would be a software element, capable of being hosted anywhere within the SD-WAN VPN. It should also be capable of being deployed in a redundant configuration, and also be capable of being redeployed with minimal disruption. Scalability, the ability to spawn instances to accommodate load, could be helpful in some situations where the SD-WAN offers more logical networking features.

In most cases, SD-WAN vendors will offer one or more Application Program Interface (API) connections to which external applications or management systems, usually part of the enterprise network operations center (NOC) can attach. These **management APIs** will always provide basic management information, but they often do not allow the external system to control all of the policies and configuration parameters of the SD-WAN. In addition, they may or may not provide access to each SD-WAN node directly. Coordinated node updates reduce the risk of inconsistent parameterization.

SD-WAN nodes can also provide an exceptionally good place to monitor services overall. The SD-WAN nodes can see both the port-side connections to users and the trunk-side connections to network services. They will have specific knowledge of interface health to the extent that the transport network connections report it, and they can even do end-to-end telemetry to assess that state of connectivity. If the SD-WAN implementation is **session-aware**, even user/application workflows may be monitored.

Management visibility could be another reason to extend the SD-WAN VPN from an extension model to an overlay model. The most critical sites and resources a company has online are usually connected via MPLS VPNs, and those resources don't typically enter the network via SD-WAN, which means that they are invisible unless they have a relationship with an SD-WAN VPN user our resource. For MSP applications, total visibility is obviously better than partial visibility, and so the management aspects of SD-WAN could well promote a more advanced SD-WAN model even without significant demand pressure from virtualization or mobility.

The MSP or network operator who sells SD-WAN won't get uniform visibility of all business traffic or endpoints if the SD-WAN VPN doesn't overlay everything, connect everything. If this channel dominates advanced SD-WAN features and sales, then that universal management story might be the deciding factor in how quickly SD-WAN moves from the basic extension model of today to an overlay model.

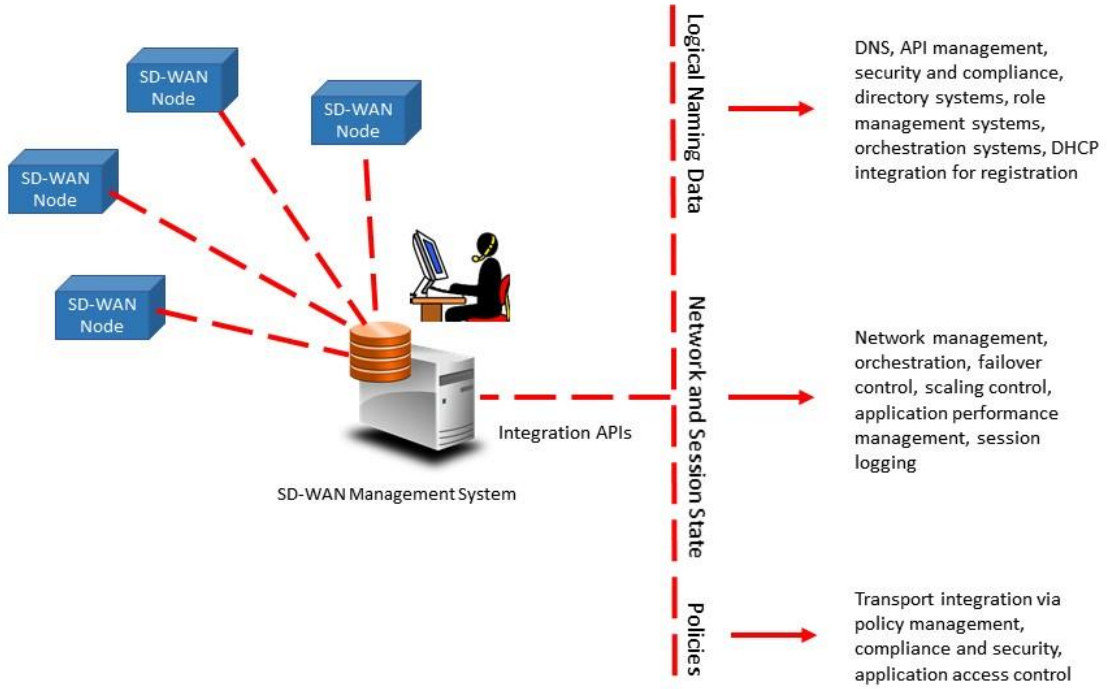


Figure 7: SD-WAN Integration

Integration of SD-WAN: Architectures and APIs

There are three kinds of API data that might be available through the SD-WAN, in most cases from the management center, as shown in Figure 7:

1. **Management data**, which would likely include the health of each node and the health of the port and trunk (transit network) interface(s) for each node. Some SD-WAN products will also make the health of sessions or connection relationships available. The latter might be per-node, per session/connection, or both. Routing table data, meaning the association between a node and the available user/resources there, as well as trunk/transit routing information is also normally available.
2. **Logical name data**, which would be the logical names assigned to users and resources, and the associated address information. This may also include information on the user's role (as part of a hierarchical name, for example), the classification of the resource (also perhaps part of the hierarchical name), and perhaps the identification of connection policies.
3. **Policy data**, which would contain all the policies that govern information movement, as well as policies relating to registration, etc. This will be highly variable depending on the implementation, but policies can control routing of information, fail-over and scaling (including load balancing), and alternate routing and QoS information.

The number of APIs that might be used for this could also vary; some vendors might expose everything through a common API and others might have different APIs for different data types. Check with the vendor for the specifics, including the kind of API used and the programming rules to attach to it.

Might applications want to draw on SD-WAN's knowledge of users and resources? Might they want to be able to directly access the SD-WAN hierarchy of logical names? We think that is very likely, and even more likely that the SD-WAN's naming/addressing tools will need to be coordinated with other sources of name/address resources already in use. Orchestration, for example, would require re-registering the addresses associated with logical names. Depending on the nature of the SD-WAN implementation, this might take place automatically, and some SD-WANs may also be able to integrate directly with the orchestration tools to provide "SDN" services.

There will almost always be an option to develop a software component to integrate an SD-WAN API data set with other applications. In some cases, the SD-WAN vendor may anticipate the need for certain kinds of integration and provide a custom interface or custom tool to provide it. Look for features to integrate the SD-WAN with orchestration tools like Kubernetes, with cloud provider tools, and also with applications that provide API directories that may have to be synchronized with resource names in the SD-WAN's logical name system.

Specific IP network tools like DNS and DHCP might be supported through vendor-provided integration too. DHCP services are usually provided by a function or server that operates within an "IP subnet", a LAN-connected community of users or data center resources. The DHCP server works from a range of subnet addresses, and anyone or anything that sends to a member of its community directs the traffic to a "default gateway" that's an onramp to that subnet. This gateway also serves as the exit point from the subnet for all traffic. If SD-WAN VPNs provide port-side registration, then the registration function could integrate with DHCP, either providing the address to the user/resource, or recording the address given.

DNS services are deployed hierarchically in most cases, meaning that a company will provide DNS services for its own URLs, and that service will pass requests for outside URLs up a hierarchy until the address can be resolved. This is a step necessary to get an IP address; the URLs themselves are not used to route traffic. Logical names might also be stored in a DNS server, and the SD-WAN logical name feature could be used to

Any DNS or DHCP services that are connected to the LAN that serves as a user or data center on-ramp to the VPN will still be accessible with SD-WAN connectivity. If either DHCP or DNS services are "deeper", meaning off the LAN and beyond the default gateway, then they'll have to be addressable to the SD-WAN node-connected users. If you are using SD-WAN in an MPLS-VPN extension mode, that typically doesn't require any changes since those servers are probably on the corporate MPLS VPN. If you are using an overlay SD-WAN model to which everything must connect, then you'll need to make sure that your DNS/DHCP services are also connected onto the SD-WAN VPN.

Note that DNS and DHCP requests are control packets, which means they don't have an explicit IP address. The way control packets are handled by SD-WAN nodes is an important thing you'll need to ask your vendor. Ideally, and likely universally in the future, SD-WAN implementations will provide for the explicit steering of control packets to a specific underlay network. Some may also provide for things like extended subnets, what are normally Level 2 domains can be spread across an enterprise, or even the cloud. This capability will require some form of address assignment mediation, and there may be a

benefit to special handling of traffic routed within a subnet, if it's not geographically confined. Again, check with your vendor.

The key evolutionary or strategic point regarding the logical networking evolution of SD-WAN is that, as SD-WAN becomes a true NaaS framework, its own user, resource, and address mappings become a logical source of the same kind of information normally provided by DHCP and DNS. We could presume that every user and resource on some future SD-WAN had an "internal URL" that represented its logical identity. Registration, as described earlier, would associate a user with both an internal URL and an SD-WAN VPN address. Registration could serve as the basis for exporting via API the DHCP functionality inherent the SD-WAN, as well as an internal source of addresses.

If you combine the registration and security features of SD-WAN with integration APIs and internally created DNS and DHCP, you can see how SD-WAN could revolutionize application access control. If users are explicitly known and registered on connection, and if application APIs are similarly known and registered, it would be easy to set policies regarding the ability of users to connect to applications, and also to define QoS priorities for those users entitled to connect. In theory, even web users could be assigned to "roles" that were then associated with application and access controls.

Perhaps the most important aspect of SD-WAN integration, and the most important mission for APIs, is the integration of SD-WAN with orchestration tools designed for deployment and redeployment of virtual machines and containers, and with service discovery tools like API brokers and managers. In a virtual world, we may always know what we want, but not always know where it is. SD-WAN, if it recognizes logical identity and maintains a link between it and SD-WAN VPN addresses, would always be able to supply the "where" to any "what".

API features are highly variable among the community of SD-WAN vendors, so add API exploration to your list of considerations. Remember that without proper API support it will be difficult to harmonize SD-WAN features relating to naming, addressing, and management with application tools that integrate workflows, orchestrate deployments, or manage access controls. You can't have too many API features.

A Summary of Important SD-WAN Features

We've made many comments on SD-WAN and its optimum features and directions. In past CIMI Corporation blogs and reports, we defined a set of what we believe are the critical features to look for in SD-WAN implementations. We'll look at each of them in this section, referring you to earlier sections where the material has already been covered, and providing an explanation of implementation and value where the feature has not already been mentioned.

1. **Session awareness**. With the possible exception of IoT events, everything in application networking is a transactional, contextual, stream of messages. In terms of traditional data networking, this is called a **session**, and sessions are relationships between transacting parties that endure over time. Because they are the real basis of communication, it makes zero sense if SD-WAN implementations aren't session-aware. Those that are not will almost certainly have to evolve to support session awareness over time. Session awareness should be coupled to name awareness, and also linked to security/encryption and prioritization, and in advanced offerings to rerouting traffic.

2. **Explicit logical-name support**. We've talked about this requirement throughout this tutorial, but most of the focus is in the section on virtualization support. Without the context of a "logical" name for users and resources, an SD-WAN implementation can't facilitate mobility, redeployment, scalability, and the other attributes we see becoming critical requirements. Capabilities in this area range from "none" to extensive hierarchical name management, and the more a vendor offers the more future-proof their solution will be.

3. **Application/Service awareness**. Effective handling of congestion, and even the control of access to resources protected by security and governance policies, demands knowing what "services" and "applications" are, not just the identity of specific components or users. This can also be linked to logical name support, creating a more powerful model.

4. **SD-WAN-as-a-Service Model**. There are multiple related definitions to this feature set. First, SD-WAN should present a uniform "network-as-a-service" model, offering branch offices, data centers, and cloud computing elements the kind of network connection they expect, with all the connection services they expect. Most users and applications expect to connect to an IP network as a member of a subnet, and to have DNS, DHCP, and perhaps things like firewall and encryption services available. That has to be true for SD-WAN too.

5. **Self-Federating**. SD-WAN, like most network technologies, is controlled by parameters and policies. Since an SD-WAN VPN is a distributed system of nodes, it's critical that the controlling policies and parameters be federated or shared among all the nodes to ensure consistent operation throughout.

6. **uCPE Support**. "Universal CPE" or uCPE is a concept evolving in the network operator equipment space to allow a generalized device on the premises to be loaded with "virtual functions" that add features like SD-WAN, firewall, etc. to a single device and

maintain them in updated form over time. This reduces the number of devices connected in series on the premises, which improves both performance and availability, and of course a single device is likely less expensive to buy and operate than multiple devices. Since SD-WAN can be uCPE-hosted, and since some uCPE features should be integrated with SD-WAN in a specific way, a vendor with some detailed capabilities in this space is valuable.

7. **Extensive APIs for Integration**. SD-WAN will always be used in conjunction with other tools to manage the network, manage name/address translation, support deployment, redeployment and scaling of application components, control application access, assign user roles, and integrate workflows across hybrid and multi-cloud configurations. All of this will require interworking between SD-WAN tools and other network and IT tools.

There are other features an SD-WAN might provide, but we believe these are the most important. The more of these features a vendor supports—and supports well—the more likely the product is to serve your current and future needs.

The Future of SD-WAN

Every technology space, whether network or IT, consumer or business, is faced with the inevitable dualistic pressure of differentiation versus commoditization. Today, something valuable and simple in the world of networking is inevitably going to get implemented in open-source form. SD-WAN features needed to support a simple extension model of SD-WAN deployment could be hosted in a router, assembled in a couple of weeks from an open-source network software inventory, or created using OpenFlow or P4. What that means is that SD-WAN is inevitably going to be driven by an obsessive need for vendor differentiation. That means that it's inevitably going to become more and more advanced in terms of features, taking us toward that goal of logical networking and NaaS.

To complicate matters, SD-WAN vendors may have competition from another space. SDN vendors are already extending their mission from the data center outward to the network at large. Unlike the early, basic, SD-WAN models that were aimed at branch connection economies, data center SDN extension focuses on applications. If you extend an application network as a goal, where do you extend it other than to the user? Everyone ends up in the same place, universal overlay connectivity.

Fortunately for SD-WAN vendors, the SDN vendors who offer SD-WAN features have been very slow to embrace the concept of "logical networking" in any form. This is likely because the early SDN players focused on multi-tenant management in the data center, and have only recently started to offer broader connectivity options, including branch networking and the "basic" SD-WAN model, and extension to the cloud. This expansion is probably due to competitive pressure in the SDN space, but because there are fewer overlay SDN players out there, market movement is likely to remain slow.

SD-WAN has already faced pressure for feature expansion beyond the basic level, with the need for support for cloud connectivity. As SD-WAN support for cloud applications expands into broader support for virtualization, as it must, SD-WAN and SDN may actually end up converging. CIMI Corporation has called the ultimate destination of both SDN and SD-WAN "logical networking". What will the new model be called by the market, and what will the convergence mean to vendors?

One possibility is that SD-WAN will be forever linked to the basic model, with its vendors doomed to commoditization and eventually death. Another is that SD-WAN will break out of the limited model to create the premier piece of the "logical networking" or "NaaS" puzzle. We're in an age where any term that gets good ink is adopted to describe anything even remotely linked to the term's original meaning.

The name we give to "logical networking" in the future may be decided by how well the various options for SD-WAN sales play against each other. Network operators and cloud providers dominate in the SDN world, and if they were to be the drivers of logical networking, then "SDN" probably wins as a name, and today's SDN vendors have the double advantage of the linkage with operators and the ownership of the winning name. MSPs and enterprises like the SD-WAN term better, and sales to either could move the SD-WAN space further, and faster, than operator sales of enhanced SDN.

Historically, network operators have been slow movers in innovation. Since SD-WAN is a competitor to MPLS VPNs, and likely to be even more competitive over time, many operators move to offer SD-WAN services only defensively, and may not be interested in pressing for aggressive new SD-WAN benefits to speed up adoption of the new technology. Enterprises and MSPs, eager to offer better services to

businesses and more than happy to take advantage of lower-cost Internet connectivity, could end up pressing for more features from SD-WAN vendors.

The marketing channel factor complicates the question of whether the evolution of SDN or SD-WAN will create "logical networking". Network operators, meaning the telcos and cable companies, are the fastest growing channel for SD-WAN sales, but their value proposition for SD-WAN is still primarily the MPLS-VPN-extension basic model. Public cloud providers are said to be looking at an "SD-WAN-as-a-Service" possibility. The cloud players not only move faster, they're more likely to be interested in logical networking and in promoting SD-WAN as a NaaS framework. Today, though, the majority of SD-WAN sales are made directly to enterprises or through an MSP.

Open-source and white boxes create yet another dynamic. The basic model of SD-WAN could be cobbled together in very little time from open-source components. It could be programmed via the P4 flow-definition language and run on white box switches too. It could even be incorporated into routers or Level 3 switches. All that in combination means that the SD-WAN vendors can't safely hunker down on the basic model. They need to move up in features, and the logical-networking path is the likely way to advance.

If there were no vendors who offered a clear pathway toward logical networking today, the concept might be little more than wishful thinking. That's not the case. While the **strategic positioning** efforts of SD-WAN vendors are hardly remarkable, it is possible to see all the pieces of logical networking falling into place. It will require some effort for prospective buyers to dig out the right details, but with persistence you can find an SD-WAN solution that will not only meet current branch/cloud requirements but also evolve into the logical networking position that CIMI Corporation believes is essential to get the best connectivity in our virtual world.

Using this Tutorial to Assess Vendors or Services

The goal of any technology selection, be it product or service, is to fulfill the requirements set by the business mission. SD-WAN is no exception, and the goal of this tutorial has been to explain SD-WAN technology in the light of its application. "Application" means not only what's needed today, but what's likely to be needed in the near term. The SD-WAN space is perhaps the fast-evolving space in all of networking, and it's evolving not because of what's being offered but because of what's needed.

The challenge for you, if you're a buyer, is that the basic model of SD-WAN will fulfill most missions today. Add in basic cloud support and the total number of fulfilled missions rises even further. All of this makes it seem that there are many different SD-WAN choices out there. In terms of the name of the technology, that's true. In terms of the application of technology to present **and future** missions, it's far less true.

We're not going to do a vendor assessment here. The space is evolving too fast, and we can't justify continually refreshing a free tutorial. Anyone who wants to look at the current state of the SD-WAN market, and its vendors, should supplement this tutorial with the reading of our blog (http://blog.cimicorp.com). However, this is a technology tutorial, and so it's looking at the market as it is and as it will develop, and the same is true for features.

That means that your mission, as a reader, is to align the features we're citing here with the offerings you're presented with. There are a few fairly simple steps involved.

Step one is to **compare your SD-WAN mission to the "basic" model**. In its basic form, SD-WAN will let you connect smaller branch sites, too small for carrier Ethernet and MPLS VPN connections, to your corporate VPN. Most current SD-WAN deals focus on this, and most are short-sighted. Do you use the cloud at all? Do you deploy applications into virtual machines or containers on premises? If you do, you are already beyond the basic model, and you should not settle for vendors who support nothing else. The need to go beyond the basics has been around for two years or more. If your vendor hasn't added cloud support, at the minimum, they're out of touch with market reality.

The basic SD-WAN support offered by vendors should allow you to extend a single MPLS VPN to remote locations using the Internet. It should also allow you to back up MPLS connections with the Internet, to use multiple ISPs, to completely replace MPLS VPNs with SD-WAN, and to extend SD-WAN across multiple MPLS VPNs with or without Internet connectivity. All these models can be depicted graphically, so get your prospective vendors to draw the pictures.

The second step, then, is to **evaluate the cloud support offered**. There are a lot of different cloud service elements, and a lot of different addressing models employed. In many cases, enterprise applications will cede front-end application processing entirely to the cloud. If that's the case, you won't need to worry too much about SD-WAN features in the cloud. However, in nearly all cases there's a critical point where work is handed off to traditional applications. That hand-off point is where you will likely need to think about SD-WAN networking.

Ideally, a collection of components inside the cloud will be allowed to continue to communicate using the cloud provider's internal addressing. Where you need to be able to intercept things with SD-WAN is

where the cloud component has to be visible to something inside the data center, on your corporate VPN, or where something in your data center or VPN is visible to the cloud. That probably means that you'll have a limited number, perhaps only one, cloud application IP address that will have to be joined to your VPN. How do you do that? Be sure that you're not dragging traffic/workflows from the cloud provider's own network, out to the Internet and perhaps even into your VPN, only to send them back. Cloud workflows need to stay in the cloud.

Multi-cloud is important, especially for enterprises widespread enough to consider SD-WAN. What works for one cloud should work for any number, so make sure that your SD-WAN product can support at least the three major public cloud providers (Amazon, Google, Microsoft). It's best if the SD-WAN vendor can supply an application note or demonstration that illustrates their approach, and as always ask them to draw diagrams.

The third step is to **consider virtualization in your own data center**. Data center virtualization using virtual machines (VMs) is common, and container-based virtualization is exploding. Virtualization success depends on orchestration of the steps needed to deploy and connect application components, which obviously includes a "virtual network" or "NaaS" requirement. Popular tools for both VM and container orchestration include plugins that enable the integration of a NaaS, but there are other ways to provide that capability as well.

Most users who make a mistake with SD-WAN will likely make it because they've not considered virtualization in their requirements. Tunnel vision focused on branch connectivity is common today, but to advance the value of SD-WAN in any dimension will require the integration of data center and application connectivity as well. Vendors who offer that in some form already have recognized the trends and taken steps to address them. They'll have an easier journey than the vendors still stuck in branch connectivity, and you'll have an easier journey as a buyer if you pick a vendor with advanced virtualization support.

The next step is to **have the vendor diagram the entire network**. Show every node, every transport network option, and clearly outline the boundaries of the SD-WAN VPN and any area of the corporate VPN where SD-WAN features will not be available. Have them lay out a logical addressing scheme if they support it, not for everything in your network but perhaps for a single application and its users. You will need to understand the SD-WAN NaaS **as a NaaS**, and without those pictures it will be difficult for you to visualize where you're going.

The final step is to **consider your source**, meaning where you're getting the SD-WAN service/product. If you're dealing with the SD-WAN product vendor directly, you can work with them to answer your questions and make necessary assurances. Information on future directions, features, etc. are more credible. If you are dealing with a reseller or service provider, you have to consider how much knowledge and leverage they'd have with the actual owner of the technology, the product, in assessing any representations that might be made of future changes or even current features. We recommend you always consult the website of the actual product vendor for authoritative information on the features, current or planned.

You'll also see, throughout this tutorial, places where we suggest strongly that you examine the specific implementations of SD-WAN being offered to you. Do that, and you'll save yourself a lot of grief.

Some Final Thoughts

All technologies these days seem to suffer from either the sin of oversimplification or the sin of overcomplication. SD-WAN is if anything a more extreme example of this polarized position. Vendor data is sometimes so basic it's hard to know if anything is being done, and we've seen sites that have obscured excellent technology in poor positioning and too much detail. Users need to know how to work something before they care about how it works.

In this tutorial, we've tried to present SD-WAN as an abstract technology wave, a transformation in network thinking that will ultimately deliver the much-hoped-for notion of network virtualization or "network-as-a-service", NaaS. That's what we believe it is, but like all technologies it's a bit of the classical "grope the elephant" problem. Vendors present features in almost the same way as a hypothetical wildlife reporter might say "if you want a snake…" or "if a tree is what you're looking for…" with respect to our elephant behind the screen.

You don't come to terms with elephants by treating their individual pieces as nothing more than what they seem to be. You don't come to terms with SD-WAN that way either. This is a new networking model, even today. In the next five years it could well become the dominant model, and that means that if you're looking at SD-WAN today you have to be thinking about your needs five years down the line. We've tried to present that future view here too.

Some people are going to be disappointed that we've not compared all SD-WAN vendors (there are more than two dozen) against the model we've described here. As this piece was being written, we received some input from vendors describing new features. The next year or so will surely bring even more change, and so specific vendor information would be quickly obsolete. This is a free, open, document and so we can't afford to spend a lot of time updating it.

Is the document valuable even without vendor specificity? We think so, but ultimately that's up to you, the prospective buyer. It's also up to you how much effort you put into using this document to help assess SD-WAN vendors and models. Knowledge is power, and we hope to add to yours in both dimensions. Thank you for your attention.