CIMI Corporation

Software-Defined WAN in the Age of Virtualization

A special market report produced by CIMI Corporation for an area we have identified as one of great strategic importance, and reviewing the SD-WAN offering of 128 Technology, the vendor who best matches our solution quality criterion.

Software-Defined WAN in the Age of Virtualization

Software-Defined WAN (SD-WAN) is an evolution of SDN technology aimed at providing virtual private network (VPN) services as an overlay, using a combination of private-network tools such as MPLS VPNs and the Internet. The original SD-WAN mission was to extend MPLS VPN coverage to sites where the user had no MPLS VPN connectivity, including small locations, international locations, home workers, and public cloud hosting points.

Early applications of SD-WAN were promoted directly to the enterprise buyer or to managed service providers (MSPs) who sold either a multi-technology/multi-geography service or an assured service based on lower-level network services. Beginning in 2017, there was increased interest in SD-WAN by traditional service providers, owing in part to the competitive risk SD-WAN posed to their traditional VPN and VLAN services, and in part to the fact that SD-WAN could be valuable as a means of creating uniform VPN services as infrastructure transitioned to Software-Defined Network (SDN) and Network Functions Virtualization (NFV).

The broad success of SD-WAN is likely linked to this recent trend of service provider interest. Network operators already hold a position of trust with prospective SD-WAN buyers, and have a strong influence on buyer technology and service planning. However, all the trust in the world can't promote something that has little or no value. Service providers need to understand what buyers will value from the SD-WAN they offer, which means understanding what implementations of SD-WAN are more valuable than others.

In our view, even the recent service-provider-centric thinking regarding the opportunities available and the features that are valuable fall short of market potential. Virtualization technology in any form, including the container/Docker revolution and public cloud computing, demands a high degree of control over the addressing of resources and the means of mediating connections to resources, from users and from application components in and out of the cloud. In our blog on the topic, we noted that this mission of address mediation is likely the "killer app" for SD-WAN, but it's an application that not all SD-WAN vendors support well.

*There are many documents, reports, and sources of information on SD-WAN. Everyone understands their basic mission of extending the scope of corporate VPNs beyond MPLS. CIMI Corporation would be adding nothing to the discussion if we did not see a gross omission in the market dialog that has emerged. That omission is the lack of a thorough exploration of the fundamental issues that virtualization raises, not some of the time but **all of the time**. We raise those issues here, identify the specific SD-WAN features that we believe are essential in fulfilling the real mission of the technology, and cite a vendor solution that fits our model.*

The Virtualization Mission for SD-WAN

The boundary point between applications and users has always been an important point in network architecture. Everything in an IP network has to be addressable, and workflows have to connect using

the addresses of each element in the flow. In virtualization, resources are elastic and assigned dynamically, moving and scaling as required. Since application elements are also dynamic, this means that the boundary between the two is almost tectonic in nature, with the same kind of possible negative consequences when things slip too much.

One particular issue that's arising at the boundary is that of addressing. Applications and services are built from components that have to be addressed in order to exchange information. For decades, we've recognized that one of the issues in IP is the fact that an address has two different meanings, and virtualization is pulling that issue to the forefront. It's the addressing of application components and resource points that creates that boundary tension noted above. If you can't make addressing work, you can't make any form of virtualization work either.

Let's suppose you're a user of an application we'll call "A1". You would likely access that application through a URL, and we'll assume the URL has the same name as the application. That's clearly a logical address, meaning that you don't care where A1 is, only that you want to get to it. When you click on the URL, a domain name server (DNS) decodes the URL name to an IP address, and this is where things go sideways. The IP address identifies A1 as a network location, because when a packet is sent to A1 it has to go somewhere specific in the network. We have transitioned, perhaps unknowingly, from a logical address reference (the URL) to a physical network reference (the IP address).

OK, you might be thinking, so what? If our IP address is in fact the address of A1, we get packets to it regardless of how we split hairs on what the IP address represents. And in the old days of fixed hosting of applications, that was indeed fine. Think now of what happens with virtualization. Let's look at a couple of scenarios.

Scenario one is that the host where A1 lived fails, and we redeploy it to another host. My conception of A1 doesn't change, but now packets have to go to a different IP address to get to A1. At the least, I have to update my DNS to reference it correctly. I may also have to tell my client systems to "poison their DNS cache", meaning stop using a saved address for A1 and go back to the DNS to get a new one. I may also have to reflect the change in A1's address in any firewall or forwarding rules that depended on the old address. In short, I might have created a bunch of issues by solving a single problem of a failed host.

Scenario two is where so many users are hammering at A1 that I decide to scale it. I now have two instances of A1, but my second instance will never get used because nobody knows about it. So, I update the DNS, you say? Sure, but in traditional DNS behavior that only redirects everything to the second instance. I add a load balancer? Sure, but that load balancer now has to be where the DNS points for everyone, and the load balancer then schedules the individual instances of A1. I have the same issues as my first scenario as far as firewalls and forwarding and DNS updates. I get the same issues again if I drop one instance and go back to my one-address-for-A1 model.

Not tired yet? OK, imagine that in either of these scenarios, I've used the public cloud as an elastic resource. I now have to expose a public cloud address of A1 through an elastic address or NAT translation of the IP address, because the cloud provider can't be using my own IP address space directly. Now not only does my IP address change, I have an outlier IP address (the one the public cloud exposed) that has to be routed to the cloud provider's gateway with me, and that has to work with firewalls and forwarding rules too.

Let's stop scenarios and talk instead of what we'd like. It would be nice if the IP address space of a business contained only logical addresses. A1 is A1 in IP address terms, whether it's in or out of a cloud, whether it's a single instance or a thousand of them, and whether it's moved physically or not. I'd like to be able to define an IP address space for my business users and their applications that would reflect access policies by address filtering. I'd like to do the same on the resource side, so my hosted components could only talk with what they're supposed to. This is the challenge of virtualization in any form, meaning the cloud, containers, VMs, SD-WAN, whatever. It's also a challenge that almost no player in any of these spaces is meeting head-on.

Mobile users pose challenges too, in a sense in the opposite direction. A mobile user that moves among cells will move away from where their IP address is pointing, meaning that traffic would be routed to where they were when the address was assigned, not to where they currently are. This problem has been addressed in mobile networks through the mobility management system and evolved packet core (MMS and EPC), which use tunnels that can be redirected to take traffic to where the user actually is without changing the address of the user. This is a remedy few like, and there's been constant interest in coming up with a better approach, even within 5G.

What approaches have been considered? Here's a high-level summary of things that might work, have worked, or maybe somebody hopes would work:

1. Address translation. Almost everyone today uses "private IP addresses" in their home. Their home networks assign addresses from a block (192.168.x.x) and these addresses are valid within the home. If something has to be directed outside, onto the Internet, it's translated into a "real" public IP address using Network Address Translation (NAT).

2. DNS coordination. In my example above, a DNS converted "A1" as a URL to an IP address. If we presumed that every resource addressable on an IP network was always addressed via a DNS, then updating the DNS record would always connect a user to a resource that moved.

3. Double Mapping. A DNS record could translate to a "logical IP address" that was then translated into a location-specific address. This has been proposed to solve the mobile user problem, since the user's "second" address could be changed as the user moved about.

4. Overlay address space. An overlay network, created by using tunnels to link "virtual routers" riding on a lower-level (presumably IP) network, could be used to create private forwarding rules that could be easily changed to accommodate moving applications and moving users. Some SD-WAN vendors already do "logical addressing". Mobile networks use MMS/EPC, as already noted, for tunnel-based user addressing.

5. Finally, the IETF has looked at the notion of using one address space (IP, in most cases) over what they call a "non-broadcast multi-access network" or NBMA, which provides a codified way of what could be called "who-has-it" routing. A network entry point contacts exit points to see who has a path to the addressed destination.

All of these approaches look at the problem from the network side, and all of them require specific networking expertise and tools to adopt. Buyers really want a solution that is packaged and architected

for easy deployment and efficient management. The best approach to doing that would seem to lie in a version of the third point above, the "double mapping". If we had a tool that could assign logical names to things, manage by logical name, and control the logical associations and workflows among VPN members, without being nailed to IP addresses, we would have a tool perfect for the virtualized world.

It's never useful to define a set of critical features with no suggestion of how they could be obtained. The good news is that the right implementation of SD-WAN technology can offer the kind of capabilities we need.

SD-WAN, Virtualization, and Address Mediation

Long ago, Internet thinkers realized that it wasn't really necessary for these two IP addresses to work the same. The user device doesn't have to **be** reached, only **responded to**. Servers know who you are because you contacted them. Thus, most Internet users have a private IP address that's their location within their home network. A network address translation (NAT) process prepends the home's IP address, assigned by your ISP, to a message you send. The web server responds to that address, which gets the response to your home gateway, which then unpacks the NATted address and gets it to you. This same process is often used within branch offices and even workgroups within offices, because it saves on scarce public IPv4 addresses. Without it, we'd have run out and gone to IPv6 long ago.

Modern container architectures are also based on assigning private IP addresses within application subnetworks, then exposing the "public ports" in the corporate VPN space. Inside a virtualized container-hosted resource pool, the actual host can be anywhere. That means that you would expose selective interfaces, the ones intended for public access, but you'd have to map them to the proper host and load-balance them if there was scaling to be done. Some of the internal interfaces, representing scalable components, would also have to be load-balanced.

In theory, we could also look at a movable user model. A "real" user might access the company VPN via any number of on-ramps, each representing a facility or area in which the user might be operating, and also perhaps including a gateway to a mobile user. We might envision a translation here—private subnet address maps to "virtual user address".

The common thread here, a thread SD-WAN might exploit, is the notion of having "local" or private addresses used as on-ramps for either applications or users. The thing they're on-ramping to, of course, is another address space. Most SD-WAN technology builds a uniform VPN address space by doing some sort of overlay on a combination of connection technologies, including MPLS VPNs and the Internet. Since SD-WAN technology can sit in all business sites and many can also be deployed as a software agent in a cloud, it would be easy for SD-WAN to create an overlay VPN that can touch all users, all applications, all resources. Most SD-WAN products do that, and little more.

If you added in some logic, you could further enhance the mapping. Users and application components could have "logical names" and these names could then be used to attach instances of the components to a load-balancer. Logical names could also map real VPN addresses to instances hosted in private IP subnets or within a cloud provider. For end users, you could assign users logical names that would link

them to a user- or role-specific subnet wherever they enter the network. Users could be associated with the applications they're allowed to use, not connected by default to every application and resource on the network.

If SD-WANs know enough to map at the logical level, they'd also know enough to do at least some rough validation of connectivity. Most workers have specific access rights to applications, but it's not convenient to manage those rights. Companies like Microsoft have grouped workers into "roles" but that doesn't always give the granularity you need. What's needed is a mechanism to associate users and applications into non-exclusive relationships that then build connectivity rights automatically.

Logical names don't displace DNS servers and URLs. Those tools are associated with the public addressing of resources. The purpose of a logical name is to provide a linkage between a resource or user and that public-link portal point. We can assume that a component knows what it is, and so can use its logical name to get an association with a private/public mapping, probably through a "logical edge" element that acts as the conduit through which applications and users access the network. If this element can provide unified addressing inside and outside the cloud, supporting application components and users alike, and offering logical name management, it can perform the critical function of address mediation.

It's the concept of a flexible edge element performing address mediation that maps so cleanly to SD-WAN. SD-WAN functionality sits at the "user/application" edge of a virtual private network. If SD-WAN agents can be deployed inside a public cloud, then cloud-hosted components can be used with the VPN seamlessly, and when SD-WAN is combined with normal IP address elasticity from the cloud provider, it can also adapt to redeployment of components. SD-WAN is also easily integrated with load-balancing features and DNS.

SD-WAN Feature Requirements for Address Mediation

The need for and value of address mediation was not recognized when SD-WANs first appeared, and most SD-WAN providers have not designed their approach to meet the new market requirement set. The specific features essential to providing address mediation in our virtual world are rarely even mentioned by vendors, much less highlighted. They are, in priority order:

1. **Session awareness**. There is no possible, rational, process for mapping and remapping the addresses of network resources that doesn't accommodate the fact that application workflows are inherently a message series with specific contextual meanings to each message. The "session" that bounds this message context has to be treated as a whole when disconnecting and reconnecting, exercising firewall protection, or doing load balancing. Absent session awareness, routine events in virtual resource configurations will create illogical sequences of messages by breaking up and reconnecting improperly.
2. **Explicit logical-name support**. If IP addresses are what a user or resource are known by, then changing them changes who the user/resource is. The use of a logical name is fundamental to address mediation because it establishes user identity outside the IP

address space, and then allows the names to map to users/resources as required by the current application deployment configuration and worker roles.

3. **Application/Service awareness**. Many SD-WAN products are designed only as user-on-ramps to a VPN that already connects host resources. They don't provide any special support of applications or application services. This renders them far less useful in conjunction with any form of virtualization or cloud computing because they can't provide any form of address mediation on the resource side. They can also offer only limited, or no, support for meaningful application prioritization, which can reduce worker quality of experience and productivity.

4. **SD-WAN-as-a-Service Model**. An appliance-based SD-WAN, or an SD-WAN that requires a specialized hosting point, is creates additional operational burdens on users or providers of SD-WAN services. Such an SD-WAN solution may also be incompatible with some public cloud computing services. A software-based solution that can be deployed on standardized servers, cloud servers (as part of a container or application image), or in edge appliances running traditional operating systems is a much more versatile approach and widens the range of users and resources that can be brought into the SD-WAN domain.

5. **Self-Federating**. A good SD-WAN solution must be able to federate all of the global policy, naming, and addressing elements through the entire SD-WAN automatically. Otherwise discrepancies in configuration, policies, and addresses will result in failures of connectivity, security violations, or governance challenges that will increase operations cost and risk.

The challenge for those considering SD-WAN is that most vendors will claim support for these areas, but few actually offer much beyond the minimal features that are intrinsic to any SD-WAN implementation. We believe that the right approach has to be architected into an SD-WAN product, not simply added to public messaging and analyst reports.

128 Technology as a Solution to an SD-WAN Virtualization-Centric Mission

The 128 Technology solution to SD-WAN is totally different from that of competitors. Instead of looking like a glorified version of Network Address Translation (NAT) or a simple implementation of overlay-modeled SDN, 128 Technology is an **advanced logical router**. Using what they call "Secure Vector Routing", they first divide their packet flows into sessions by recognizing the start and end of each session within that flow. These sessions are then "routed" at the session level, ensuring that everything that belongs to a single logical flow is handled in the same way.

SVR creates a stateful router overlay on top of whatever the network's underlying technologies might be. That means that SVR can integrate stateful firewall and load balancing capability easily; they have already done the difficult part by identifying inherently stateful flows. SVR also supports the application

of handling policies for performance management and capacity/resource control **at the logical session level**, so that all the packets linked to a given mission are handled in the same way. Since SVR policies are exchanged across all the 128 Technology SD-WAN endpoints, they're consistently applied everywhere. This approach provides all the "traditional" SD-WAN features, but in a better way, a way aligned with the emerging issues of virtualization.

SVR routers are "routers" in a real sense; they can handle transit traffic, perform peer exchanges to align their reachability and route information, and be hosted where convenient. There is nothing to prevent a service provider deploying 128 Technology SVRs from placing some in "transit" locations on a VPN rather than only at points of user/service access. This lets network operators who offer SD-WAN provide some differentiation over MSP-overlay-type competitors.

The next level of the 128 Technology solution is **service-oriented networking**. Networks support many different applications, and applications have many different kinds of information flows. To simplify the process of handling flows, 128 Technology allows administrators of the SD-WAN to define any number of discrete **services**, each of which has specific requirements and tolerance for QoS variables. Each service is then associated with specific handling policies. The policies can influence priority scheduling, and where there are multiple network path options available at a given endpoint, the specific network service on which the traffic rides. Applications can then map to services, eliminating the need to define policies for every different application being run.

Service-oriented networking also means that the various network paths can be assigned a traffic-type quota based on service definitions, to prevent rerouting of traffic from interfering with the QoE of critical applications. Administrators can override automatic policies, but because the policy control of traffic is unusually precise, it's unlikely they'd need to do that.

Services are hosted, of course, and the place where services resides is a "tenant" in 128 Technology documentation. A tenant can contain users and services, and the tenant names can be qualified to facilitate fine-grained policy control. If there is an "Accounting" tenant associated with a set of services, then the users within that tenancy will have routes to those services. A subset tenancy like "Accounting.Compliance" would have all the services of the overlying tenancy and also any specific services assigned to the qualified Compliance tenancy. Where a given user or group of users doesn't have a given service within their tenancy, they have no route to those services, which means that access control is inherent. It is possible for the administrator to override the automatic access control facilities using explicit access lists.

IP networks provide permissive connectivity by default; 128 Technology SVRs create a network that is zero-access by default. All tenant/service routes have to be explicitly created, which means that all connectivity has to be authorized. Management tools make it easy to audit the relationships between services and tenants for compliance monitoring.

All these names have to be managed, of course, and 128 Technology uses Qualified Service Names (QSNs). QSNs are in a very limited way similar to URLs in that they are logical names that have to be translated, but a QSN can be applied to both resources/services and users, and can refer to a single instance or multiple instances of either. An administrator can assign as many QSNs as they like, and associate as many services with a given QSN as business operations require. QSNs and service names

are what policies reference, which makes them portable across the real IP addresses that users or resources might have.

These QSNs are enormously helpful in organizing a complex visualization-dependent IP VPN. Just as traditional routers exchange reachability data when they're peered, SVR routers exchange their QSNs and service information. That means that it isn't necessary for administrators to explicitly build routes, or to rebuild them when something changes. This capability is especially important when an SD-WAN is built over a combination of different underlying network services, using the same or different technologies and providers. It allows for control of the critical bridging of routes across the network mix.

For cloud and virtualized-hosting environments, 128 Technology adopts a software-agent model of implementation, suitable for any x86 hardware platform. The agents can be composed with application units of deployment to integrate the SD-WAN VPN with all virtualization- or cloud-based elements, and control of the routes makes sure that service traffic doesn't use a charged resource (as many public clouds are) to bridge between underlying networks. This model is integrated with the 128 Technology pricing model to align SD-WAN pricing with the feature value to the buyer/user.

For locations where access bandwidth is limited, 128 Technology has the advantage of using no additional headers on packets, which can increase available link bandwidth by as much as 20% in comparison with other technologies where extensive headers are used. The underlying networks, and all the 128 Technology SVR interfaces, then operate at higher throughput. The overhead benefit will likely benefit every SD-WAN application where some sites have a limited range of access options, particularly in rural or international locations.

All of this is controlled by a multiplanar control/management structure, the top of which is the 128T Conductor that provides centralized name and policy management for all the SVRs. The next layer, the 128T Control layer, is sustained by the peer relationships among the SVR elements, and it also collects all the statistics and provides the analytics needed to support routing decisions and enforce policies. This sits on top of the Session-Oriented Data Plane, which is formed by the combination of the Software Line Card Engine (SLICE) complement of the SVRs. SLICE is where high-speed packet/flow classification and policy enforcement is handled.

Basic SD-WAN features are available in many forms, from many vendors. Some of these vendors, recognizing the changes in the market and the need for differentiation, have added portions of what we believe to be the critical SD-WAN feature set for the virtualization-driven future of software and services. In those cases, the limited original product scope is of the vendors' offerings are easily visible in the documentation, which references primarily supported network connection types rather than address-management features. Compare the web collateral of other vendors with that of 128 Technology and you'll see what we mean.

128 Technology is exactly the opposite of its competitors in its approach. Even though it offers all the usual SD-WAN benefits, it explains its capabilities in a top-down way, developing its capabilities by linking them to user needs and benefits. It's clear that's because the product architecture was done from the top as well. The 128 Technology approach to SD-WAN aligns precisely with the future pathway that virtualization and cloud computing is taking, and that path is one every business will take over time.

