White Paper

Data Center Overlay Technologies

What You Will Learn

In the modern data center, traditional technologies are limiting the speed, flexibility, scalability, and manageability of application deployments. There is emerging interest in the industry in overlay technologies, which may address some of these challenges. This document examines why the industry is moving toward adoption of network overlay technologies, and it describes the benefits and challenges of these technologies.

Introduction

Successful businesses rely on quick development of new applications, which requires data centers to be more efficient, scalable, and agile. The industry has been seeking ways to use virtualization technologies to offer these benefits, not only for computing and storage resources, but also for the network infrastructure.

The adoption of server virtualization has been rapidly increasing throughout the industry, which has benefited data center administration as a whole, allowing greater flexibility and agility in the provisioning and placement of computing workloads. However, network connectivity has not kept pace with such innovations in the computing environment, still offering a rigid approach to provisioning transport services. Network overlays may help address this challenge.

Requirements of the Modern Data Center

Modern data center fabrics must meet certain requirements to accelerate application deployment and meet develop and operations (DevOps) needs.

Cloud Integration

Many enterprises today use virtual private clouds or hybrid clouds in which some part of the workload is moved to the public cloud. In such environments, the provider or administrator may want to offer elastic services, so that individual tenants can expand, reduce, or move workloads throughout their service-offering lifecycle. The data center fabric used by both service providers and enterprises must be suitable for this model.

Mobility of Virtual Devices

One of the benefits of server virtualization is the capability to support live migration of virtual machines as they move from one physical server to another, without requiring the virtual machines to shut down and restart. This move may be triggered, for example, by workload rebalancing policy or scheduled maintenance. In such moves, the virtual machine must retain adequate information about the network state, such as the IP address and MAC address. Essentially, the address of the end host should be independent of its location in the network.

Scaling of Forwarding Tables

With the increased adoption of virtualized servers in the modern data center, additional scaling demands are being placed on traditional network devices. Because these devices are still using end-host information (IP address and MAC address) to make forwarding decisions, this state information needs to be propagated to the entire data center fabric's forwarding tables. This propagation may lead to dramatically increased scale, especially in large-scale multitenant environments, in which multiple instances of end-host information must be installed and propagated throughout the fabric.

Scaling of Network Segments

In today's data centers, VLANs are used extensively to segment the network into smaller domains to enable traffic management, secure segmentation, and performance isolation for services across different tenants. The VLAN construct is a tool of the 1990s that us reaching the end of its usefulness. VLANs were designed to scope broadcast domains, and they have been used extensively to concatenate servers and various network services. However, VLANs are subject to scalability limitations resulting from space limitations (only 4000 VLANs are allowed) and from control-plane limitations.

Coupling of Physical and Logical Connectivity

Administrators need to be able to deploy and expand workloads anywhere in the data center, yet still maintain constructs such as IP addresses and broadcast domains (VLANs) where these new services are being deployed. Maintaining these constructs can be accomplished by extending the VLAN domain over a larger area, but this approach may affect the availability of the network by increasing the size of the fault domain, and it requires considerable administrative overhead and reconfiguration, which may introduce errors or misconfiguration. Ultimately, the Layer 2 network needs to be expanded without affecting the availability of existing services.

Coupling of Infrastructure and Policy

In today's data centers, it is common practice to group entities with like membership into smaller segments (VLANs) to provide a way to identify, segment, and enforce policies between such groups. Likewise, IP addressing schemes may be classified with the same subnet boundaries. This tight coupling of network policy and network infrastructure is a cause of many of the inefficiencies and limitations that are found in data centers today, because a change in policy often results in a change in topology, and a change in topology often results in a change in policy. A mechanism is needed that allows these independent constructs to be decoupled from one another so that the deployment of services in the data center can be managed separately from the network addressing and topology.

Virtualized Networks

As data centers consolidate multiple tenants onto a single shared environment, individual tenants, instead of the overall fabric administrator or provider, may need to manage address space. At times, tenants' address spaces across these virtual networks may overlap. Additionally, and more fundamentally, individual tenant address spaces must be managed independently from those of the infrastructure or provider to help ensure that any changes in infrastructure or tenant addresses do not affect each other. Therefore, the data center fabric must allow per-tenant addressing that is separate from addressing by other tenants and also separate from the infrastructure.

Optimized Forwarding

Today's networks vary in their forwarding efficiency depending on the underlying protocol being deployed. In Layer 2 networks, most deployments depend on variations of the Spanning Tree Protocol to eliminate loops by blocking redundant paths. However, this protocol often leads too much wasted capacity in scaled-out environments. Although Layer 3 networks can use multipathing, they are tuned to make forwarding decisions based on shortest-path mechanisms for specific destinations. In many instances, the desired path may not be the shortest path to the destination: for instance, when traffic from a given source may need to transit a service such as a load balancer or firewall that is not on the shortest path to the destination.

Additionally, sometimes multiple shortest-paths may be available. This may be the case, for instance, when two or more external routers are exiting the data center or virtual network. If movement of a virtual machine is involved, the closest exit router may change; however, because the IP forwarding does not discriminate between devices that are all one hop away, selecting the optimal path for forwarding is difficult, potentially leading to "trombone" forwarding effects.

Reduction in Dependency on Traditional Protocols

A challenge that always arises when extending Layer 2 networks is how can a solution meet all the preceding requirements and at the same time avoid dependencies on traditional protocols that are not scalable, are prone to configuration errors, and have far-reaching failure domains? One example of such a protocol is the Spanning Tree Protocol, which offers limited redundancy for Layer 2 networks, has limited scalability due to its requirement to eliminate data-plane forwarding over redundant paths, and is prone to misconfiguration and other errors than can lead to catastrophic network failure.

Introducing Network Overlays

Although the network overlay concept is not new, network overlays have gained interest in the past few years because of their potential to address some of the requirements mentioned in the preceding section. They have also gained interest with the introduction of new encapsulation frame formats purpose-built for the data center, including Virtual Extensible LAN (VXLAN), Network Virtualization Using Generic Routing Encapsulation (NVGRE), Transparent Interconnection of Lots of Links (TRILL), and Location/Identifier Separation Protocol (LISP). Network overlays are virtual networks of interconnected nodes that share an underlying physical network, allowing deployment of applications that require specific network topologies without the need to modify the underlying network. This section examines the advantages and disadvantages of overlays.

Benefits of Network Overlays

Network overlays offer a number of benefits that can help meet some of the challenges of the modern data center.

Optimized Device Functions

Overlay networks allow the separation (and specialization) of device functions based on where a device is being used in the network. An edge or leaf device can optimize its functions and all its relevant protocols based on end-state information and scale, and a core or spine device can optimize its functions and protocols based on link-state updates, optimizing on fast convergence. This approach also reduces complexity for network devices. In the case of server-based overlays, this function is implemented on the server. In the case of network-based overlays, this function is implemented on the first switch (at the top of the rack).

Fabric Scalability and Flexibility

Overlay technologies allow the network to scale by focusing scaling on the network overlay edge devices. With overlays used at the fabric edge, the spine and core devices are freed from the need to add end-host information to their forwarding tables. Additionally, the placement of end hosts is more flexible because the overlay virtual network no longer needs to be constrained to a single physical location.

Arbitrary Layer 2 Connectivity without Layer 2 Underlay

Another benefit of network overlay technologies is that they can decouple the network service provided to end hosts from the technology used in the physical network. For example, Layer 3 routed networks may be run pervasively throughout the data center; however, by using certain network overlay technologies, Layer 2 services also can be extended across a routed topology.

Overlapping Addressing

Most overlay technologies used in the data center allow virtual network IDs to uniquely scope and identify individual private networks. This scoping allows potential overlap in MAC and IP addresses between tenants. The overlay encapsulation also allows the underlying infrastructure address space to be administered separately from the tenant address space.

Separation of Roles and Responsibilities

With the encapsulation used in overlay technologies, separation of administration domains can also be achieved. The administrator of the fabric (infrastructure) can be responsible for fabric addressing, availability, and load balancing, and the individual tenants can be responsible for their own addressing policies and services without affecting infrastructure policies.

Overlay Network Use Cases

Overlay networks can be deployed in private, public, and hybrid cloud environments in the data center to support the following use cases:

- Simplified management: Use a single point of management to provide network resources for multitenant clouds without the need to change the physical network.
- Multitenancy at scale: Provide scalable Layer 2 networks for a multitenant cloud that extends beyond 4000 VLANs. This capability is very important for private and public cloud hosted environments.
- Workload-anywhere capability (mobility and reachability): Optimally use server resources by placing the workload anywhere and moving the workload anywhere in the server farm as needed.
- Forwarding-topology flexibility: Add arbitrary forwarding topologies on top of a fixed routed underlay topology.

Challenges of Overlays

New technologies solve problems but also bring new challenges. This section describes some of the challenges of overlays.

Decreased Fabric Visibility

The adoption of overlay technologies may decrease the visibility of the fabric as a whole because network constructs that exist in the overlay network are hidden from the underlay fabric. For example, traceroute in the overlay will not report individual underlay hop counts.

Troubleshooting Complexity

The presence of overlays with virtual topologies makes troubleshooting more complicated because the network administrator must investigate the mapping of the virtual topology on top of the physical topology.

Network Overlay Technologies in the Data Center

Overlay networks can be classified into either of two categories:

- Network-based overlay networks
- Host-based overlay networks

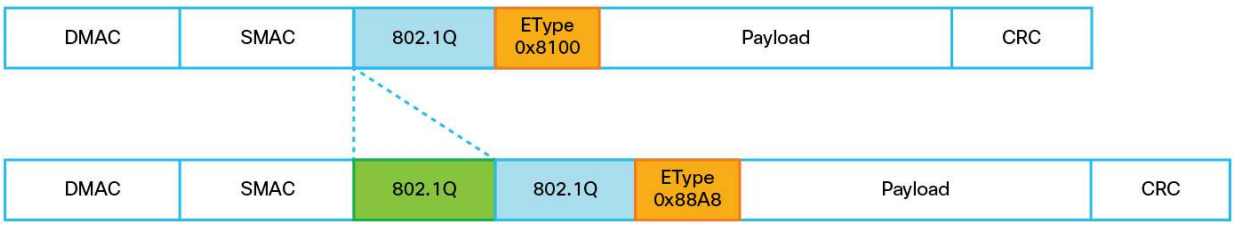
Network-Based Overlay Networks

Network-based overlay networks has been around for many years to address various challenges in data center networks, metropolitan area networks (MANs), and WANs.

IEEE 802.1ad Provider Bridging or IEEE 802.1q Tunneling

Also known as IEEE 802.1QinQ or simply Q-in-Q, provider bridging is a tunneling specification that allows multiple VLAN headers to be inserted into a single frame initially used for Metro Ethernet networks. Stacking the 4-byte VLAN tags (for which 12 bits are allocated for the VLAN ID) allows customers to administer their own VLANs (C-TAG) within a service provider's allocated VLAN (S-TAG), potentially allowing over 16 million segments with two tags (Figure 1).

Figure 1. IEEE 802.1ad Provider Bridge Frame Format

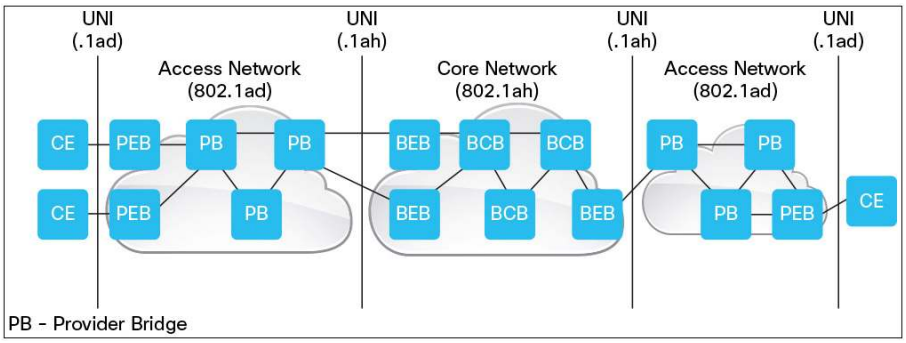


No additional control protocols are required for Q-in-Q tunneling other than those already in use for standard Ethernet bridging such as Multiple Spanning Tree (MST) Protocol or Rapid Spanning Tree Protocol (RSTP). As a result, only the devices performing encapsulation and de-encapsulation of the additional IEEE 802.1Q tag need to be aware of this function, so that tags are applied and mapped correctly. All other interim devices in the core of the tunneled network do not require knowledge of the embedded C-TAG (Figure 2).

Although Q-in-Q tunneling does provide a way to scale the virtual network segment space from 4000 to more than 16 million addresses, it does not provide a solution to hide the MAC addresses from the core of the (provider) network. As a result, all (customer) MAC addresses will be learned everywhere across the entire domain by all devices, potentially affecting the scalability of the solution.

Provider bridging and IEEE 802.1q tunneling have experienced significant deployment in Metro Ethernet environments, but only limited adoption in data centers.

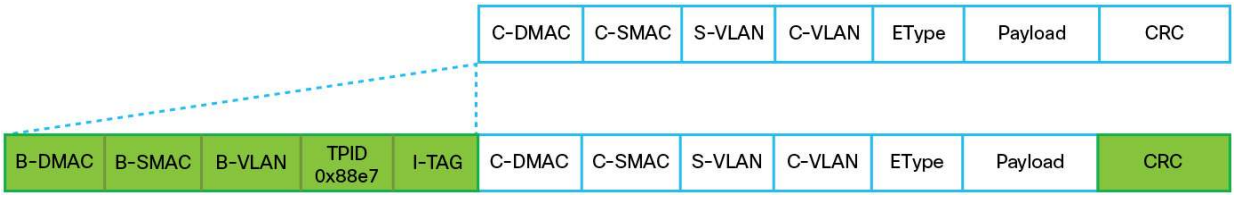
Figure 2. Deployment of IEEE 802.1ad Provider Bridge and IEEE 802.1ah Provider Backbone Bridge



IEEE 802.1ah Provider Backbone Bridges or Mac-in-Mac Tunnels

IEEE 802.1ah, or provider backbone bridge (PBB), encapsulates end-user or customer traffic in the provider's MAC address header, allowing the backbone edge bridge (BEB) to support large numbers of service instances, and at the same time allowing customer MAC addresses to be hidden from the backbone core bridge (BCB). The PBB employs MAC address tunneling encapsulation to tunnel customer Ethernet frames across the PBB network, a backbone VLAN ID (B-VLAN) to segregate the backbone into broadcast domains, and a new 24-bit backbone service instance identifier (I-SID) is used to associate a given customer's MAC address frame to the provider's service instance (Figure 3).

Figure 3. IEEE 802.1ah Provider Backbone Bridge Frame Format



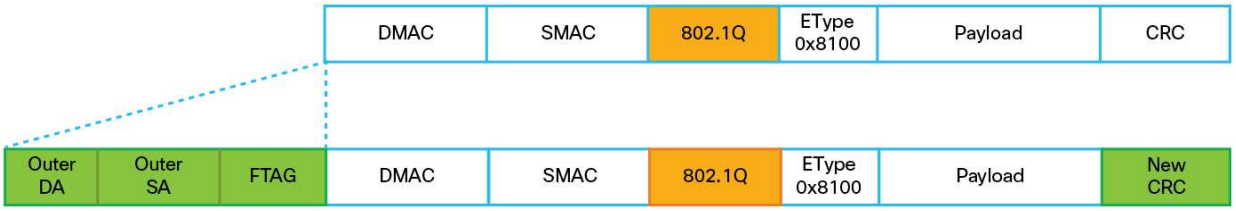
In addition to capabilities specified in IEEE 802.1ad, PBB can hide customer MAC addresses from the provider network through the additional MAC-in-MAC encapsulation; however, it faces challenges with features that many provider networks want such as multipathing, traffic engineering, and carrier-class resiliency because it still relies on Spanning Tree Protocols for loop avoidance.

Cisco FabricPath

Cisco® FabricPath switching allows multipath networking at Layer 2 and encapsulates the entire Layer 2 frame with a new Cisco FabricPath header. Cisco FabricPath links are point to point, and devices encapsulate frames at the ingress edge port of the Cisco FabricPath network and de-encapsulate frames on the egress edge port of the Cisco FabricPath network. This new encapsulation allows the core of the Cisco FabricPath network to be hidden (through overlay technology) from the host state information, reducing the scaling requirements of Cisco FabricPath core devices.

Because Cisco FabricPath encompasses the entire Ethernet frame in a new encapsulation, all nodes on the Cisco FabricPath network need to support Cisco FabricPath to look up and forward the frame throughout the rest of the network (Figure 4).

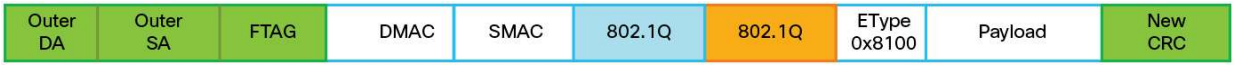
Figure 4. Cisco FabricPath Frame Format



Cisco FabricPath also introduces an additional tag called the forwarding tag (FTAG), which can be used to describe and segment multiple forwarding topologies, by mapping Ethernet VLANs to a given topology at the Cisco FabricPath edge. The frame is encapsulated with the appropriate FTAG as it is forwarded throughout the Cisco FabricPath network, where forwarding is constrained to a given topology.

Although the current deployment of Cisco FabricPath does not support extension of the segment space beyond 4000 VLANs, at the time of this writing some applications of Cisco FabricPath can extend the 12-bit VLAN ID to a 24-bit segment ID, by appending an additional IEEE 802.1Q tag, allowing over 16 million segments (Figure 5).

Figure 5. Cisco FabricPath Frame Format with Additional IEEE 802.1Q Tag for Increased Segment Space



Cisco FabricPath uses extensions to the Intermediate System-to-Intermediate System (IS-IS) protocol to exchange unicast and multicast location and reachability information and to forward traffic in the network using Cisco FabricPath headers. Because IS-IS is a dynamic link-state routing protocol, it can detect changes in the network topology and calculate loop-free routes to all other nodes in the network, with each node having a complete link-state database that describes the state of the entire network.

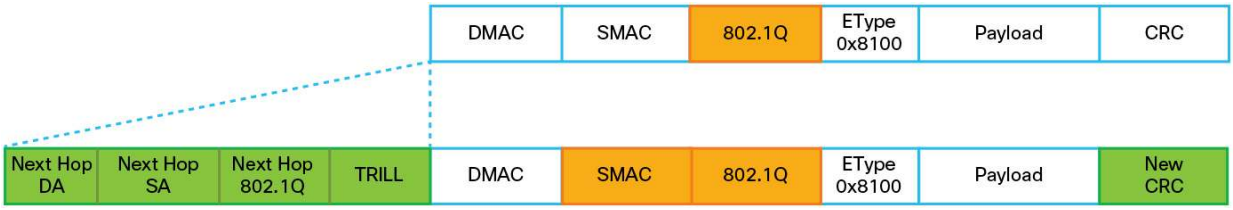
Cisco FabricPath learns end-host device information through regular data-plane learning, but an additional enhancement called conversational MAC address learning is also available to specific devices with Cisco FabricPath enabled. In conventional IEEE 802.3 bridges, MAC addresses are learned on an interface based on the source MAC address received, regardless of the bidirectional nature of communication flows. This approach may lead to exhaustion of MAC address tables, which could lead to flooding in the network. With conversational MAC address learning, an interface learns the source MAC address of an ingress frame only if that particular interface already has a destination MAC address present in the MAC address table. If the source MAC address interface does not already know the destination MAC address, that MAC address is not learned.

Cisco FabricPath is supported on the Cisco Nexus® Family of switches and is experiencing increased deployment for intra-data center fabric connectivity.

Transparent Interconnection of Lots of Links

IETF Transparent Interconnection of Lots of Links, or TRILL, is similar in many ways to Cisco FabricPath and is also a Layer 2 multipathing technology. It is implemented by devices called routing bridges (RBridges) and adds a new encapsulation to the frame. However, this encapsulation is implemented in such a way that it is compatible and can incrementally replace existing IEEE 802.3 Ethernet bridges. With the encapsulation of a new Ethernet MAC address header, the original MAC address header is left unmodified and hence can pass through intermediate Ethernet bridges. However, as with Cisco FabricPath, the new encapsulation also allows the core of the TRILL network to be freed from having to learn edge-host addresses (Figure 6).

Figure 6. IETF TRILL Routing Bridges Frame Format



Another difference between TRILL and Cisco FabricPath is the forwarding path. RBridges are similar to routers in that when a TRILL frame requires forwarding by an intermediate RBridge, the outer Layer 2 header is replaced at each RBridge hop with an appropriate Layer 2 header for the next hop, and a hop count in the TRILL header is decremented. Despite this, the original encapsulated frame is preserved, including any VLAN tags.

Similar to Cisco FabricPath, TRILL uses extensions to IS-IS as its routing protocol. The link-state protocol provides enough information between the RBridges so that they can compute pair-wise optimal paths for unicast traffic and calculate distribution trees for multidestination frames.

End-host address information can be learned either through standard data-path source address learning or through the optional End-Station Address Distribution Information (ESADI) protocol. ESADI frames are encapsulated as regular TRILL data frames so that participation is optional. If an RBridge does not implement the ESADI protocol, it does not de-capsulate or processes the frames, but instead forwards the frames as if they were regular multicast TRILL data frames.

As with Cisco FabricPath deployments today, TRILL currently has no provision for extending the segment space beyond 4000 segments.

IEEE 802.1aq: Shortest-Path Bridging

Shortest-Path Bridging (SPB) is defined in IEEE 802.1aq and is targeted as a replacement for Spanning Tree Protocol, which blocks traffic on all but one alternative path. It is a Layer 2 multipathing technology that allows all paths to be active with multiple equal-cost paths, providing fast convergence times, and it can support larger segment spaces to accommodate scalable virtual networks. Similar to both Cisco FabricPath and TRILL, SPB uses extensions to IS-IS as a link-state routing protocol to calculate the shortest-path tree (SPT) and discover the topology of the network.

SPB supports two modes of operation (and hence two different encapsulations), and both modes can coexist in the same network. These modes are Shortest Path Bridging-VLAN ID (SPBV) and Shortest-Path Bridging-MAC Address (SPBM). Both modes use the link-state IS-IS protocol to calculate the SPT and multicast distribution tree (MDT).

Shortest-Path Bridging-VID

SPBV is very flexible and can be used in networks implementing IEEE 802.1Q VLANs, IEEE 802.1ad provider bridges (discussed earlier), and IEEE 802.1ah provider backbone bridges (discussed earlier). However, SPBV uses the VLAN ID in these encapsulations to perform service delineation and load balancing. SPBV also differs from SPBM in that MAC addresses are learned on all bridges that lie on the shortest path.

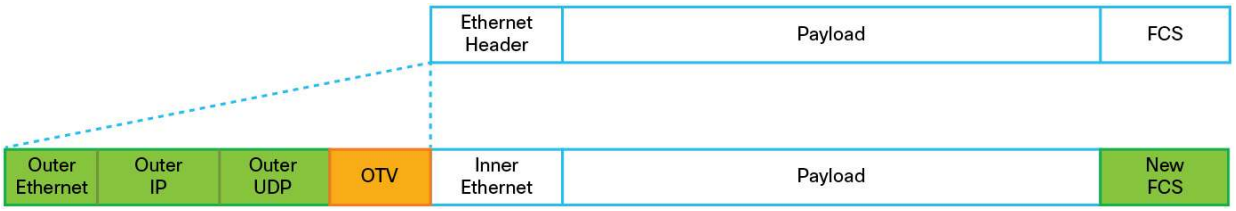
Shortest-Path Bridging-MAC

SPBM specifically uses IEEE 802.1ah provider backbone bridge frame formats for data-plane encapsulation. Unlike SPBV, SPBM uses I-SIDs (I-TAG) for service delineation, but for load balancing VLANs can also be used. For forwarding, SPBM uses a combination of one or more B-VIDs, known as backbone-MAC (B-MAC) addresses that have been advertised in IS-IS. Additionally, in SPBM edge MAC addresses are never learned or looked up in the core of a IEEE 802.1aq network; B-MAC addresses are distributed through the control plane through IS-IS, thus eliminating B-MAC address learning in PBB.

Overlay Transport Virtualization

Cisco Overlay Transport Virtualization (OTV) is a Layer 2-over-Layer 3 encapsulation “MAC-in-IP” technology that is designed to extend the reach of Layer 2 domains across data center pods, domains, and sites. It uses stateless tunnels to encapsulate Layer 2 frames in the IP header and does not require the creation or maintenance of fixed stateful tunnels. OTV encapsulates the entire Ethernet frame in an IP and User Datagram Protocol (IP/UDP) header, so that the provider or core network is transparent to the services offered by OTV (Figure 7).

Figure 7. Overlay Transport Virtualization Frame Format



In an OTV network, the OTV edge device is responsible for encapsulation and de-encapsulation of the OTV header and IP header and exists primarily on physical switches or routers. At the time of this writing, OTV is available on the Cisco Nexus 7000 Series Switches and Cisco ASR 1000 and 9000 Series Aggregation Services Routers.

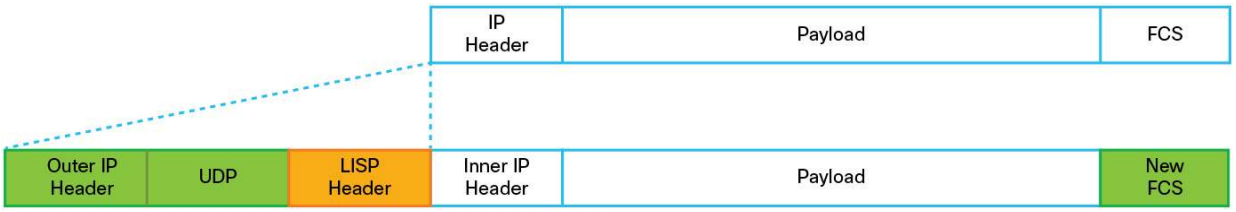
OTV uses the IS-IS control protocol to advertise reachability of MAC address or end-host information, instead of using traditional “flood and learn” techniques. This approach is especially important when OTV is used in WAN or multisite deployments, in which excessive flooding may be detrimental to the performance of the WAN. For OTV to discover other edge devices with which to peer and exchange reachability information, OTV can use multicast support enabled in the core of the network (transport network), or if multicast support is not possible, an OTV adjacency server can be used to distribute a list of all peer edge devices in the overlay.

The OTV header also offers an Instance ID field that the OTV edge device can use to select a logical table to be used for lookup by the edge device at a remote site. This feature may be useful for mapping overlapping VLAN ranges across different tenants.

Locator/Identifier Separation Protocol

The Cisco Location/Identifier Separation Protocol, or LISP, is designed to address the challenges of using a single address field for both device identification and topology location. This challenge is evident in modern data centers, where the mobility of endpoints should not result in a change in the end-host addressing, but simply the location of the end host. LISP addresses the problem by uniquely identifying two different number sets: routing locators (RLOCs), which describe the topology and location of attachment points and hence are used to forward traffic, and endpoint identifiers (EIDs), which are used to address end hosts separate from the topology of the network (Figure 8).

Figure 8. LISP Frame Format



LISP defines the capabilities and functions of routers and switches to exchange information to map EIDs to RLOCs, as well as a mechanism that allows LISP routers to encapsulate IP-based EIDs for forwarding across an IP fabric or the Internet using RLOC addresses. The devices performing the encapsulation and de-encapsulation of LISP headers are called ingress tunnel routers (ITRs) and egress tunnel routers (ETRs), respectively. LISP is currently defined as a Layer 3 overlay scheme over a Layer 3 network, and it encompasses IPv4 and IPv6 for both the underlay and the overlay.

Similar to other encapsulation schemes described previously, LISP provides a mechanism to help ensure virtual segment isolation through the addition of a 24-bit instance ID field in the LISP header, allowing more than 16 million virtual segments to be instantiated; this mechanism is set by the ITR.

Multiprotocol Label Switching

Multiprotocol Label Switching (MPLS) has been used extensively in service provider environments and even certain enterprise environments. The flexibility of the protocol and its inherent scalability, however, has renewed interest in the technology in the modern data center. Instead of forwarding traffic based on addresses in a routing table, MPLS forwarding devices forward traffic based on path labels, identifying paths in the network instead of endpoints. Additionally, MPLS supports the capability to stack multiple tags, enabling overlay services to be applied transparently from the transport network, including Asynchronous Transfer Mode (ATM), Frame Relay, SONET, Ethernet, and VPN services.

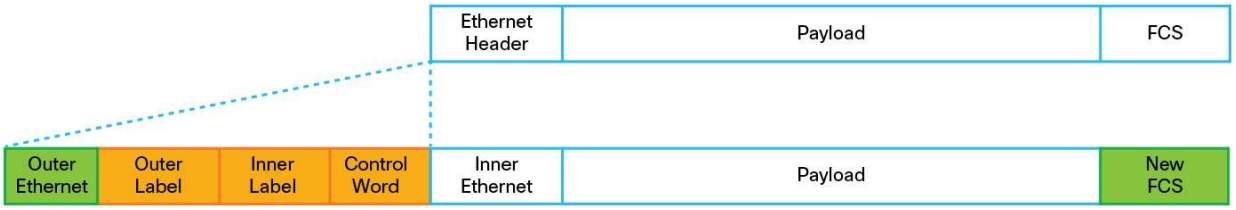
Of particular interest in the modern data center are MPLS VPNs. Through the use of label stacking; a given label in the stack can be dedicated to uniquely identifying a given virtual segment. There are fundamentally two implementations of MPLS VPNs: Virtual Private LAN Service (VPLS) and Virtual Private Routed Network (VPRN)

Virtual Private LAN Service

VPLS, as defined in RFC 4761 and RFC 4762, allows the creation of pseudowires that emulate LAN segments (for an Ethernet switch) for a given set of users, and that are fully capable of learning and forwarding Ethernet MAC addresses that are closed to that set of users. However, multiple VPLS services can be supported from a single provider-edge device. In contrast to other Layer 2 tunneling protocols (such as Layer 2 Tunneling Protocol Version 3 [L2TPv3]), VPLS allows any-to-any (multipoint) connectivity and is typically deployed in a provider network to emulate a switch or a bridge to connect customer LAN segments to create a single bridged LAN.

For encapsulation, VPLS uses MPLS labels to identify the path and virtual customer segment or VPLS instances. It uses a two-label stack, with the outer label used for normal MPLS forwarding in the provider's network, and the inner label used to determine the relevant VPLS instance. As a result, the core of the provider network that supports a VPLS service must support MPLS transport through to the provider-edge devices (Figure 9).

Figure 9. VPLS Frame Format



VPLS can be used in conjunction with IEEE 802.1Q or 802.1ad, with the VLAN tag used to identify specific virtual customer segments. Additionally, VPLS can operate in two modes that providers can offer: tag mode and raw mode.

For label distribution, discovery, and signaling, two control-plane methods have been widely adopted throughout the industry. One of the use of the Border Gateway Protocol (BGP) as defined in RFC 4761, and the other is the use of the Label Distribution Protocol (LDP) as defined in RFC 4762.

Raw mode assumes that all customer payloads are carried over the pseudowires intact, and if an IEEE 802.1Q VLAN tag is present, it is ignored. This mode does not delimit service.

Tagged mode assumes that at least one IEEE 802.1Q VLAN tag is present, and that this VLAN tag can be used to identify the customer segment. Therefore, the outer VLAN tag is removed and mapped to the inner VPLS label. On egress, the labels are removed, and the VLAN tag is appended back onto the frame for forwarding to any connected bridges. This mode delimits service.

VPLS has typically been deployed in Carrier and Metro Ethernet service offerings, though it has been used in some data center Interconnect (DCI) deployments in specific geographic segments.

Virtual Private Routed Network

VPRN, also known as BGP/MPLS IP-VPN, as specified in RFC 4364 describes a method for which a provider can use an IP backbone to provide IP VPN services for its customers or tenants. This method is routed, with individual customer VPNs routed to the provider-edge VPN instances, and it is private, with each VPN maintaining its own routing table space so that customers with overlapping routes can be supported over the same shared infrastructure.

From a data-plane encapsulation standpoint, VPRN uses MPLS labels to identify paths from a given provider-edge device to other provider-edge devices in the network, but it also uses another MPLS label to uniquely identify each customer's VPN. Before the customer's data packet is transported across the provider network, it is encapsulated with the MPLS label that corresponds to its VPN, and then the packet is further encapsulated with another label so that it is tunneled to the correct provider-edge router. As a result, the backbone provider routers do not need to know the VPN routes.

From a control-plane perspective, the intermediate nodes in the backbone use LDP to propagate path reachability information, but then they use BGP to distribute VPN routes, each of which is tagged with an MPLS label for that route, so that routing information can also be uniquely constrained to the given VPN instance (called the Virtual Routing and Forwarding [VRF] instance). As a result of this requirement, multiprotocol extensions have been added to BGP to allow it to carry a route distinguisher in addition to the prefix when sending updates. This feature helps ensure that if the same address is used in different VPNs, a single protocol instance can be used to carry different routes for different VPNs, without the need to run a separate protocol instance for each VPN.

VPRNs have been used extensively in both service provider and enterprise environments. Specifically in the data center, the edge or border node is responsible for terminating the intra-data center virtualization technology of choice (IEEE 802.1Q, IEEE 802.1ad, IEEE 802.1ah, VXLAN etc.) and mapping or routing each virtual segment to an IP VPN instance for external services.

Host-Based Overlay Network

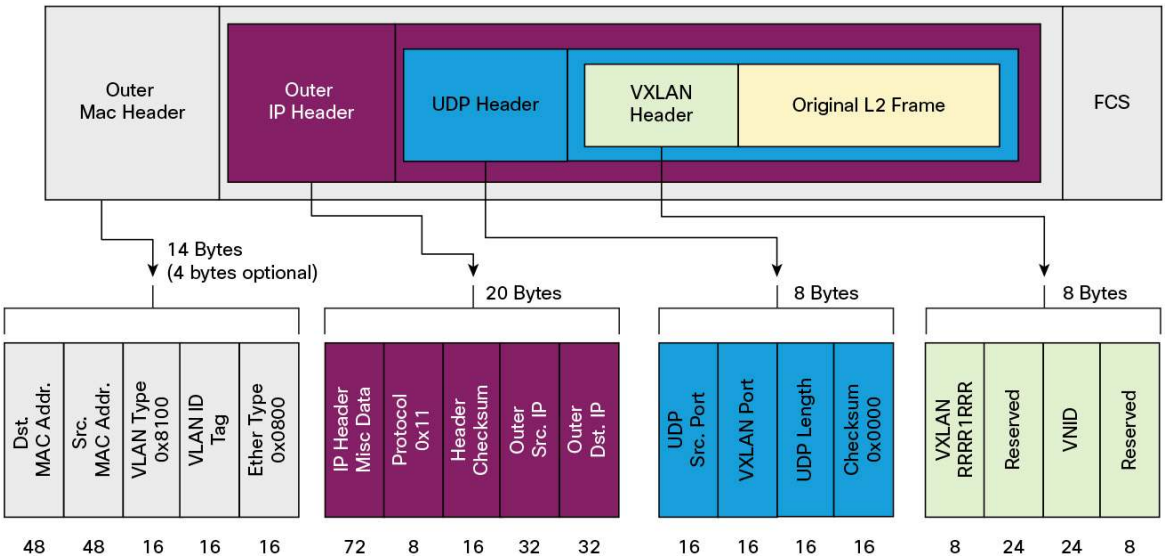
Network-based overlay networks have been around for many years and address many of the networking challenges and problems; however, three main problems remained unresolved, which the host-based overlay network was developed to address:

- Workload placement anywhere (mobility)
- Simplified and automated workload provisioning
- Multitenancy at scale

Virtual Extensible LAN

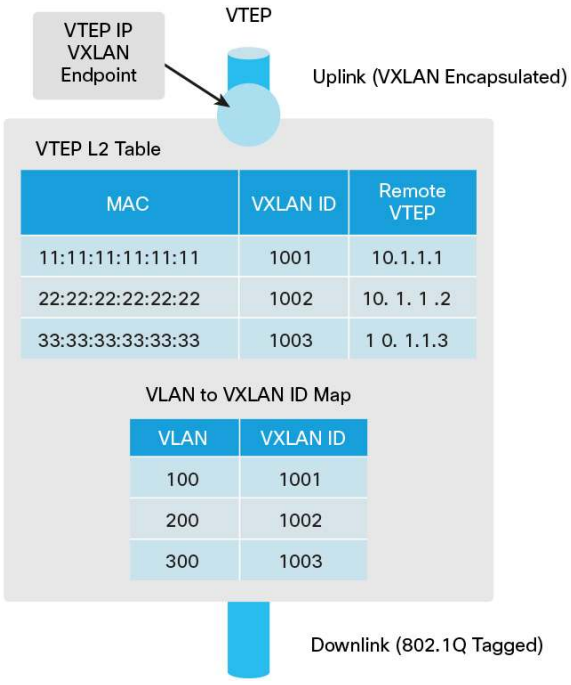
Virtual Extensible LAN, or VXLAN, is a Layer 2 overlay scheme over a Layer 3 network. It uses an IP/UDP encapsulation so that the provider or core network does not need to be aware of any additional services that VXLAN is offering. A 24-bit VXLAN segment ID or VXLAN network identifier (VNI) is included in the encapsulation to provide up to 16 million VXLAN segments for traffic isolation and segmentation, in contrast to the 4000 segments achievable with VLANs. Each of these segments represents a unique Layer 2 broadcast domain and can be administered in such a way that it can uniquely identify a given tenant's address space or subnet (Figure 10).

Figure 10. VXLAN Frame Format



With the imposed encapsulation, and similar to OTV as discussed earlier, VXLAN can be considered a stateless tunneling mechanism, with each frame encapsulated or de-encapsulated at the VXLAN tunnel endpoint (VTEP) according to a set of rules. A VTEP has two logical interfaces: an uplink and a downlink (Figure 11).

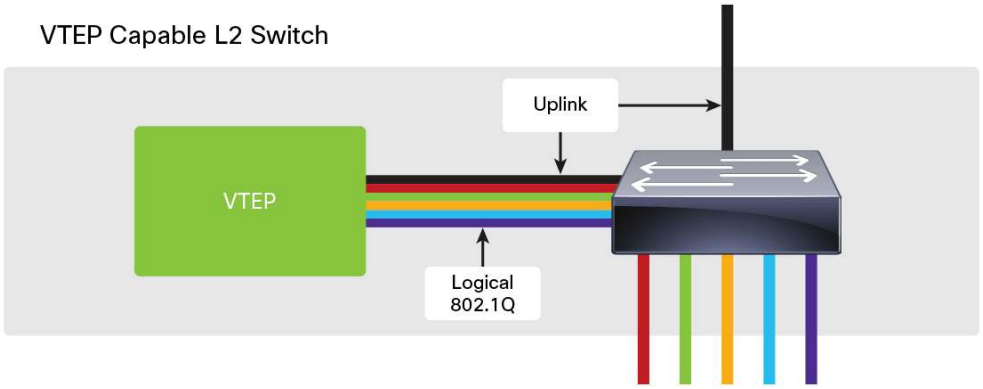
Figure 11. VTEP Logical Interfaces



The uplink is responsible for receiving VXLAN frames and acts as a tunnel endpoint with an IP address used for routing VXLAN encapsulated frames. These IP addresses are infrastructure addresses and are separate from the tenant IP addresses for the nodes that use the VXLAN fabric. The VTEP can be located either on a physical switch or within the hypervisor virtual switch in a server virtualization deployment.

VXLAN frames are sent to the IP address assigned to the destination VTEP; this IP address is placed in the outer IP destination address packet. The IP address of the VTEP sending the frame resides in the outer IP source address packet. Packets received on the uplink are mapped from the VXLAN ID to a VLAN, and the Ethernet frame payload is sent as an IEEE 802.1Q Ethernet frame on the downlink. During this process, the inner source MAC address and VXLAN ID are learned in a local table. Packets received on the downlink are mapped to a VXLAN ID using the VLAN of the frame. A lookup is then performed in the VTEP Layer 2 table using the VXLAN ID and destination MAC address; this lookup provides the IP address of the destination VTEP. The frame is then encapsulated and sent out the uplink interface (Figure 12).

Figure 12. Logical View of VTEP Switch



VTEPs are designed to be implemented as logical devices on a Layer 2 switch. The Layer 2 switch (which is usually a top-of-the-rack [ToR] switch) connects to the VTEP through a logical IEEE 802.1Q VLAN trunk. This trunk contains a VXLAN infrastructure VLAN in addition to the production VLANs. The infrastructure VLAN is used to carry VXLAN encapsulated traffic to the VXLAN fabric. The only member interfaces on this VLAN are the VTEP's logical connection to the bridge and the uplink to the VXLAN fabric. This interface is the uplink described earlier, and the logical IEEE 802.1Q trunk is the downlink.

Basically, the Ethernet frame sent by a VXLAN-connected device is encapsulated in an IP/UDP packet. The most important point to note is that the frame can be carried by any IP-capable device. The only time added intelligence is required in a device is at the VTEP, or network bridge, which performs the encapsulation and de-encapsulation. Therefore, although benefits can be gained by adding VXLAN capabilities elsewhere, doing so is not required.

The VXLAN draft standard does not mandate a control protocol for discovery or learning. It offers suggestions for both control-plane source learning (push model) and central directory-based lookup (pull model). At the time of this writing, most implementations depend on a flood-and-learn mechanism to learn the reachability information for end hosts. In this model, VXLAN establishes point-to-multipoint tunnels to all VTEPs on the same segment as the originating VTEP to forward unknown and multidestination traffic across the fabric. This forwarding is accomplished by associating a multicast group for each segment, and hence it requires the underlying fabric to support IP multicast routing.

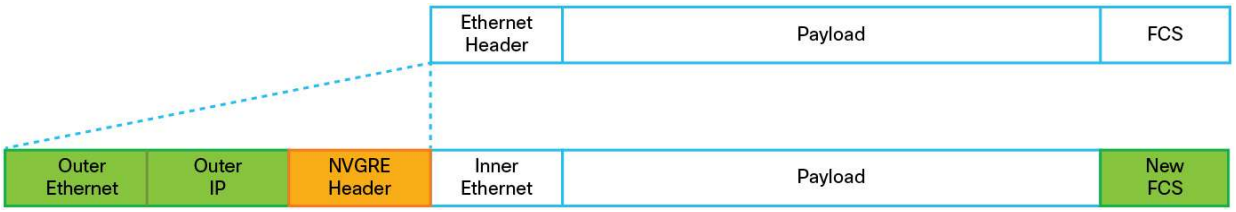
VXLAN is one of the most popular Layer 2-over-Layer 3 overlay mechanisms that are being investigated for future deployments. This popularity is mainly due to the flexibility of services (Layer 2 or Layer 3 overlay with Layer 3 IP underlay) and the availability of this encapsulation in software and hardware implementations from a variety of vendors. One barrier to deployment, however, is that multicast routing must be implemented in the underlay network to support multidestination traffic.

In summary, VXLAN is a network overlay technology design for data center networks. It provides massively increased scalability over VLAN IDs alone while allowing Layer 2 adjacency over Layer 3 networks. The VXLAN VTEP can be implemented in both virtual and physical switches, allowing the virtual network to map to physical resources and network services. VXLAN currently has wide support and adoption in switching application-specific integrated circuit (ASIC) and network interface card (NIC) hardware as well as in virtualization software.

Network Virtualization Using Generic Routing Encapsulation

Network Virtualization Using Generic Routing Encapsulation, or NVGRE, allows the creation of virtual Layer 2 topologies on top of a physical Layer 3 network. This design is achieved by tunneling Ethernet frames inside an IP packet over a physical network. Similar to VXLAN, NVGRE supports a 24-bit segment ID or virtual subnet identifier (VSID), providing up to 16 million virtual segments that can uniquely identify a given tenant's segment or address space (Figure 13).

Figure 13. NVGRE Frame Format



The NVGRE endpoints are responsible for the addition or removal of the NVGRE encapsulation and can exist on a network device or a physical server. NVGRE endpoints perform functions similar to those performed by VTEPs in a VXLAN environment, and they are also responsible for applying any Layer 2 semantics and for applying isolation policies based on the VSID.

A main difference between VXLAN and NVGRE is that the NVGRE header includes an optional flow ID field. In multipathing deployments, network routers and switches that can parse this header can use this field together with the VSID to add flow-based entropy, although this feature requires additional hardware capabilities.

As with VXLAN, the NVGRE draft standard does not specify a method for discovering endpoint reachability. Rather, it suggests that this information can be provisioned through a management plane or obtained through a combination of control-plane distribution or data-plane learning approaches.

Stateless Transport Tunneling

Stateless transport tunneling (STT) is an overlay encapsulation scheme over Layer 3 networks that use a TCP-like header within the IP header. The use of TCP fields has been proposed to provide backward compatibility with existing implementations of NICs to enable offload logic, and hence STT is specifically useful for deployments that are target end systems (such as virtual switches on physical servers). Note that, as the name implies, the TCP fields do not use any TCP connection state.

One area that STT specifically addresses is the size mismatch between Ethernet frames and the maximum transmission unit (MTU) supported by the underlying physical network. Most end-host operating systems today set the MTU at a small size so that the entire frame plus any additional (overlay) encapsulations can be transported over the physical network. This setting may result in a potential performance degradation and additional overhead compared to frames that can be transmitted with their desired maximum segment size (MSS). STT seeks to exploit the TCP segmentation offload (TSO) capabilities built into many NICs today to allow frame fragmentation with appropriate TCP, IP, and MAC address headers, and also the reassembly of these segments on the receive side.

Similar to other encapsulations discussed earlier, STT contains a virtual network identifier that is used to forward the frame to the correct virtualized network context. This identifier is contained in a 64-bit context ID field and has a larger space to address a variety of service models and allow future expansion.

Host-based overlay networks address many of the challenges posed by rigid underlay networks and their associated protocols (Spanning Tree Protocol, etc.,), but the overlay network needs to be integrated with the physical network.

A major and unfounded assumption about host-based overlay networks is that the underlying network is extremely reliable and trustworthy. However, an overlay network tunnel has no state in the physical network, and the physical network does not have any awareness of the overlay network flow. A feedback loop is needed from the physical network and virtual overlay network to gain end-to-end visibility into applications for performance monitoring and troubleshooting.

Comparison of Network Overlay Technologies

Table 1 provides a comparison of the network overlay technologies.

	VXLAN	STT	NVGRE	LISP: Layer 2
Encapsulation	• Uses UDP-based encapsulation • Uses UDP port 8472 • Adds an 8-byte VXLAN header • Encapsulates IP and non-IP Ethernet frames	• Uses TCP-based encapsulation • Adds an 8-byte STT header • Encapsulates IP and non-IP Ethernet frames • Uses nonstandard stateless TCP	• Uses GRE-based encapsulation • Uses GRE protocol type 0x6558 (transparent Ethernet bridging) • NVGRE encapsulates untagged IP and non-IP Ethernet frames	• Uses UDP-based encapsulation • Uses UDP port 4341 • Adds an 8-byte LISP header
Overlay identification	24-bit virtual network ID (VNI)	64-bit context ID	24-bit virtual subnet identifier (VSID), plus an optional 8-bit flow ID	24-bit LISP instance ID
Encapsulation overhead	50 bytes	76 bytes	42 bytes	50 bytes
Maximum size of encapsulated data payload	• Network MTU: 50 bytes • Size depends on virtual NIC (vNIC) MTU in the virtual machine, system jumbo MTU in the virtual switch (vSwitch), MTU in uplinks, and so on	• 64 KB • Large packets are segmented in the NIC (TCP segmentation), depending on the MTU of the underlying physical network • Requires reassembly at destination (performed by the receiving NIC) • Same source port must be used for all segments of a single STT frame	• Network MTU: 42 bytes • Size depends on vNIC MTU in the virtual machine, system jumbo MTU in the vSwitch, MTU in uplinks, and so on	• Network MTU: 50 bytes • Size depends on vNIC MTU in the virtual machine, system jumbo MTU in the vSwitch, MTU in uplinks, and so on
Fragmentation after encapsulation	Cisco VXLAN deployment guide indicates that network MTU should be increased 50 bytes to avoid fragmentation of VXLAN packets	None; STT uses the interface MTU and TCP segmentation	Draft RFC proposes using path MTU discovery and setting the DF bit on the outer header to avoid fragmentation after encapsulation (RFC 2003, Section 5.1)	• LISP Layer 3 draft RFC proposes two methods for handling LISP packets that exceed MTU: stateless and stateful • These methods are applied at the ITR before encapsulating
Fragmentation of encapsulated data	No information in draft RFC	None; payload size limit is 64 KB	No information in draft RFC	See above

	VXLAN	STT	NVGRE	LISP: Layer 2
Forwarding of Layer 2 broadcast, multicast, and unknown unicast traffic	• Encapsulation uses IP multicast as destination IP • Each VNI is mapped to a multicast group • Multiple VNIs can share the same multicast group	• Draft RFC leaves open the method to use • One option mentioned is to encapsulate IP multicast as destination IP, if supported by the underlay • Ingress replication can also be used, based on information obtained through control plane	• Encapsulation uses IP multicast as destination IP • Each VSID is mapped to a multicast group • Multiple VSIDs can share the same multicast group.	Draft LISP Layer 2 RFC provides two options: • Ingress replication • Use of underlay multicast trees
Equal-Cost Multipathing (ECMP) and PortChannel load balancing in underlay	• Source UDP port used by the VXLAN encapsulation is determined from a hash of the inner headers • Underlay network should use 5-tuple-based hashing	• Source TCP port used by the STT encapsulation is determined from a hash of the inner headers • Underlay network should use 5-tuple-based hashing	• Draft RFC proposes use of the 32 bits of VSID plus the flow ID for ECMP purposes • Hashing based on GRE header is not common in current hardware switches	• Source UDP port used by the LISP encapsulation is determined from a hash of the inner headers • Underlay network should use 5-tuple-based hashing
Forwarding of Layer 2 broadcast, multicast, and unknown unicast traffic	• Encapsulation uses IP multicast as destination IP • Each VNI is mapped to a multicast group • Multiple VNIs can share the same multicast group	• Draft RFC leaves open the method to use • One option mentioned is to encapsulate use of IP multicast as destination IP, if supported by the underlay • Ingress replication can also be used, based on information obtained through control plane	• Encapsulation uses IP multicast as destination IP • Each VSID is mapped to a multicast group • Multiple VSIDs can share the same multicast group	Draft LISP Layer 2 RFC provides two options: • Ingress replication • Use of underlay multicast trees
Address learning and control plane	Draft RFC provides the option for using either: • Learning and flooding approach: that is, data-plane-based learning; details about this option are provided in the draft RFC • Separate control plane (central directory with pull or push model)	• Not specified in the draft RFC; leaves open the choice of control plane, keeping it separate from the data plane encapsulation • Nicira's control plane is based on OpenFlow	Draft RFC provides the option to use any mechanism to distribute location and VSID information: data plane learning, control-plane based, etc.	LISP mapping system, supporting encoding of instance ID and MAC address 2-tuple
Quality-of-service (QoS) handling	• Nothing specified in the draft RFC • On the Cisco Nexus 1000V Switch, the uniform model is currently applied: ◦ The class-of-service (CoS) setting from the inner packet is copied to the outer header. ◦ If the encapsulated packet is IP, the Differentiated Services Code Point (DSCP) setting from the inner header is also copied to the outer header. • This is a default behavior; not configurable	Draft RFC includes two references to handling QoS settings in a tunneling protocol: • Reference to RFC 2983 for mapping DSCP from inner to outer header; 2 models can be used: uniform and pipe • Reference to RFC 6040 for handling ECN settings	Nothing specified in the draft RFC	• LISP Layer 3 specifies that inner type-of-service (ToS) field should be copied to the outer header • Explicit Congestion Notification (ECN) bits must be copied from inner to outer header • LISP Layer 2 does not mention anything about QoS parameters yet
Offload to NIC	No	Yes; uses TCP segmentation offload (TSO) and large receive offload (LRO) capabilities that are common on the NICs	No	No

	VXLAN	STT	NVGRE	LISP: Layer 2
Virtual switch support	Cisco Nexus 1000V and VMware DVS	Nicira, which is based on Open vSwitch	Microsoft Hyper-V virtual switch	None; LISP Layer 2 support is on Cisco Nexus 1000V roadmap
Scalability	1 Million Hosts	1000 hosts (Nicira claim)	Unknown	Unknown
Vendors	Cisco, VMware, Arista, Brocade, Citrix, Red Hat, and Broadcom	VMware and Broadcom	Microsoft, Arista, Emulex, Huawei, and HP	Cisco
Support in hardware switches	Arista 7150 (VTEP in hardware) and Brocade ADX	None yet	None yet	None for LISP Layer 2
Gateway from overlay to physical network	VMware vShield, Cisco ASA for Nexus 1000V Series Switch, Cisco Cloud Services Router (CSR) 1000V Series, Arista 7150 switch, and Brocade ADX	Nicira appliance	Unknown	None yet
Service insertion	Cisco vPath in Cisco Nexus 1000V	Unknown	Unknown	Unknown
Specifications	http://tools.ietf.org/html/draft-mahalingam-dutt-dcops-vxlan-02	http://tools.ietf.org/html/draft-davie-stt-02	http://tools.ietf.org/html/draft-sridharan-virtualization-nvgre-01	http://tools.ietf.org/html/draft-smith-lisp-layer2-01

For More Information

I-D.mahalingam-dutt-dcops-vxlan

- Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and Wright, C., "VXLAN: A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks," draft-mahalingam-dutt-dcops-vxlan-02 (work in progress), August 2012

I-D.narten-nvo3-overlay-problem-statement

- Narten, T., Black, D., Dutt, D., Fang, L., Gray, E., Kreeger, L., Napierala, M., and Sridhavan, M., "Problem Statement: Overlays for Network Virtualization," draft-narten-nvo3-overlay-problem-statement-04 (work in progress), August 2012

I-D.sridharan-virtualization-nvgre

- Sridhavan, M., Greenberg, A., Venkataramaiah, N., Wang, Y., Duda, K., Ganga, I., Lin, G., Pearson, M., Thaler, P., and Tumuluri, C., "NVGRE: Network Virtualization Using Generic Routing Encapsulation," draft-sridharan-virtualization-nvgre-01 (work in progress), July 2012

I-D.davie-stt

Davie, B., Gross, J., "A Stateless Transport Tunneling Protocol for Network Virtualization (STT)," draft-davie-stt-02 (work in progress), August 2012

I-D.ietf-lisp

- Farinacci, D., Fuller, V., Meyer, D., and Lewis, D., "Locator/ID Separation Protocol (LISP)," draft-ietf-lisp-24 (work in progress), November 2012

I-D.hasmit-otv

- Grover, H., Rao, D., Farinacci, D., and Moreno, V., "Overlay Transport Virtualization," draft-hasmit-otv-3 (work in progress), January 2012

rfc.6325

- Perlman, R., Eastlake 3rd, D., Dutt, D., Gai, S., and Gahnwani, A., “Routing Bridges (RBridges): Base Protocol Specification),” rfc6325, July 2011

rfc.5364

- Rosen, E., and Rekhter, Y., “BGP/MPLS IP Virtual Private Networks (VPNs),” rfc4364, February 2006

rfc.4761

- Kompella, K., and Rekhter, Y., “Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling,” rfc4761, January 2007

rfc.4762

- Lasserre, M., and Kompella, V., “Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling,” rfc4762, January 2007

rfc.6329

- Fedyk, D., Ashwood-Smith, P., Allan, D., Bragg, N., and Unbehagen, P., “IS-IS Extensions Supporting IEEE 802.1aq Shortest Path Bridging,” rfc6329, April 2012

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV Amsterdam,
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at www.cisco.com/go/offices.

Printed in USA C11-730116-00 11/13