A Survey of Network Isolation Solutions for Multi-Tenant Data Centers

Valentin del Piccolo, Ahmed Amamou, Kamel Haddadou, Guy Pujolle

A Survey of network isolation solutions for multi-tenant data centers

Valentin Del Piccolo*, Ahmed Amamou†, Kamel Haddadou†, and Guy Pujolle‡

Abstract

The Infrastructure-as-a-Service (IaaS) model is one of the fastest growing opportunities for cloud-based service providers. It provides an environment that reduces operating and capital expenses while increasing agility and reliability of critical information systems. In this multitenancy environment, cloud-based service providers are challenged with providing a secure isolation service combining different vertical segments, such as financial or public services, while nevertheless meeting industry standards and legal compliance requirements within their data centers. In order to achieve this, new solutions are being designed and proposed to provide traffic isolation for a large numbers of tenants and their resulting traffic volumes.

This paper highlights key challenges that cloud-based service providers might encounter while providing multi-tenant environments. It also succinctly describes some key solutions for providing simultaneous tenant and network isolation, as well as highlights their respective advantages and disadvantages. We begin with Generic Routing Encapsulation (GRE) introduced in 1994 in "RFC 1701", and will conclude with today's latest solutions. We detail fifteen of the newest architectures and then compare their complexities, the overhead they induce, their VM migration abilities, their resilience, their scalability, and their multi data center capacities. This paper is intended for, but not limited to, cloud-based service providers who want to deploy the most appropriate isolation solution for their needs, taking into consideration their existing network infrastructure. This survey provides details and comparisons of various proposals while also highlighting possible guidelines for future research on issues pertaining to the design of new network isolation architectures.

1 Introduction

Data centers are being increasingly used by both corporations and individuals. For example, in Cisco's forecast [1], personal content locker services, like Amazon Cloud Drive, Microsoft SkyDrive, and Google Drive, are expected to increase their total traffic by 57% in Cumulative Annual GRowth (CAGR) between 2013 (2 Exabytes) and 2018 (19 Exabytes). In addition to a growing use of data centers made by consumers (individuals), the use of data centers made by corporations will also increase.

As stated in [1], in 2013 global data center traffic reached 3.1 zettabytes for the year and is expected to grow to 8.6 zettabytes in 2018, representing a 3-fold increase. However, it is important to make a distinction between two data center types.

The first type of data center is the traditional one, which possesses specialized servers. On the contrary, the second type of data center is the cloud data center, which possesses non-specialized servers. These different data center types will not see the same increase in traffic. The traffic from traditional data centers will "only" increase by 8% CAGR between 2013 and 2018, while cloud data center traffic will see an increase of 32% CAGR during the same period , as predicted in [1]. In other words, in 2013 cloud data center workloads represented 54% of total data center workloads, and in 2018 it will represent 76% of the total data center workloads. We can therefore see a shift in favor of cloud data center.

This can be explained by one major advantage of a cloud data center over a traditional data center. A cloud data center is more prone to virtualization than a traditional data center. Indeed, cloud data centers are data centers with virtualized devices. With hardware improvement of data center nodes, it is possible to run several virtual machines (VMs) on one physical node.

Using several VMs on a physical node allows using it to the fullest of its capabilities. It therefore spends less time in an idle state and wastes less energy. Consequently, it is therefore cost-effective for both the infrastructure provider and their customers. The infrastructure provider, owner of the data center, has nodes actively processing the data of its clients instead of being held in an idle state. This increases the load time of the nodes, which in turn increases their cost-effectiveness. The infrastructure provider therefore needs fewer physical devices for a fixed number of clients. Instead of having multiple physical devices for one client, the provider has multiple VMs for this client. All these VMs can be on one physical device (Figure 1).

Virtualization also adds functionalities such as :

- Remote OS install
- Access to server console
- Reboot of frozen server.

*Valentin Del Piccolo is with the Research and Development Department of GANDI SAS, Paris, France, and a Phd student at the University Pierre et Marie Curie (UPMC), Paris, France. e-mail: valentin.d.p@gandi.net

†Dr Ahmed Amamou and Dr Kamel Haddadou are with the Research and Development Department of GANDI SAS, Paris, France, e-mail: ahmed@gandi.net, kamel@gandi.net

‡Pr Guy Pujolle is a Professor at the University Pierre et Marie Curie, Paris, France. e-mail: Guy.Pujolle@lip6.fr

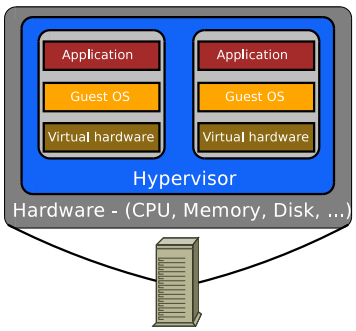


Figure 1: Two VMs on one node

- Guest OS choice.
- Possibility of server snapshots for backups.
- Hardware upgrade without shutting down the VM.
- Possibility of VM migration on a newer server with the backup image of the VM.

However, sharing a physical node among several clients implies that there is no device or data isolation. Nevertheless, clients do not want their data exposed to other clients, who might even be competitors. In order to solve this problem, it is necessary to deploy techniques that will provide client isolation. This results in clients only seeing VMs and traffic that they own, and make them believe that they are alone on the network.

The remainder of the survey is organized as follows. After quickly reviewing terminology and definitions that pertain to multitenancy isolation in the cloud (Section 2) and explaining tunneling and virtual network notions (Section 3), we detail in Section 4 some network isolation solutions developed before the cloud era in order to show why new solutions are needed for cloud data center with multi-tenant issues. In Section 5 we present those new solutions which provide multi-tenant isolation in cloud data centers. Then we focus on fifteen solutions that provide tenants traffic isolation as follows: The Locator/Identifier Separation Protocol (LISP) [2], Network Virtualization using Generic Routing Encapsulation (NVGRE) [3], Stateless Transport Tunneling Protocol (STT) [4], 802.1ad or QinQ [5], 802.1ah or mac-in-mac [6], Virtual eXtensible Local Area Network (VXLAN) [7] Diverter [8], Portland [9], Secure Elastic Cloud Computing (SEC2) [10], BlueShield [11], VSITE [12], NetLord [13], Virtual Network over TRILL (VNT) [14], VL2 [15], Distributed Overlay Virtual nEtwork (DOVE) [16, 17]. We compare them using six criteria in Section 7. We then discuss the future of tenant isolation (Section 8) and, finally, present our conclusions (Section 9).

2 Terminology

In this section we define both terms Tenant and Multitenancy. To do so, we use the definitions given in [18, 19, 20, 21].

2.1 Tenant

In Cisco Virtual Multi-Tenant Data Center 2.0 [19] a tenant has two definitions. In the private cloud model a tenant is defined as "a department or business unit, such as engineering or human resources". In the public cloud model a tenant is "an individual consumer, an organization within an enterprise, or an enterprise subscribing to the public cloud services". In version 2.2 of Cisco Virtual Multi-Tenant Data Center [20] the difference between public or private cloud has been removed and a tenant is "an user community with some level of shared affinity". To explain this definition, examples are provided in which a tenant may be a business unit, department, or work group.

Juniper gives a different definition, they state in their white paper [21] that "a cloud service tenant share a resource with a community". In order to express this definition more clearly, the example of building tenants is given. In this metaphor, a building tenant has to share the building's infrastructure just like a cloud service tenant. However a tenant can also have tenants such as stated in [21]. The given example is the case of Second Life which is a tenant of Amazon Web Services and which has tenants of its own, who could also have tenants and so on. Wider than Cisco's definition of a tenant, Juniper defines a tenant as a cloud service user. This user can be a person or a company or, as in Cisco's definition, a business unit from a company.

In this paper we use the term tenant as defined by Juniper.

2.2 Multitenancy

For Cisco [20], "virtualized multi-tenancy" is a key concept which refers to "the logical isolation of shared virtual compute, storage, and network resources".

In continuation with the building metaphor, most of the time there is not only one tenant in a building. Therefore the building is a multitenancy environment. Each tenant wants privacy so they are isolated in apartments. This metaphor is well presented in "The Force.com Multitenant Architecture" [22] and is reproduced below :

> "Multitenancy is the fundamental technology that clouds use to share IT resources cost-efficiently and securely. Just like in an apartment building - in which many tenants cost-efficiently share the common infrastructure of the building but have walls and doors that give them privacy from other tenants - a cloud uses multitenancy technology to share IT resources securely among multiple applications and tenants (businesses, organizations, etc.) that use the cloud."

For Juniper [21], multitenancy is the idea of many tenants sharing resources. It is also a key element for cloud computing. However multitenancy also depends on the service provided. For example, in an IaaS environment, the provider provides infrastructure resources

like hardware and data storage to the tenants who in turn must share them. In a SaaS environment, tenants use the same applications, so there is a chance that their data is stored in a single database by the service provider. There are security constraints to apply at each layer.

In this survey we focus on data center architectures providing tenants' traffic isolation.

3 Background

This section explains the notions of virtual networks (Section 3.1) and tunneling (Section 3.2). We also detail the relation between multitenancy, virtual networks, and tunneling in Section 3.3.

3.1 Virtual network

Microsoft defines, in [23], a virtual network as a configurable network overlay. The devices from one virtual network can reach each other but those outside of it can not, thus providing isolation to the devices inside the virtual network.

A more concise definition of a virtual network is given in [24]: "[...] a virtual network is a subset of the underlying physical network resources."

Using both definitions we see that a virtual network is a configurable overlay network that uses resources, virtual nodes, and virtual links of a physical infrastructure while at the same time keeping them isolated. A virtual node is a logical node using at most all the resources of a physical node. A virtual link works the same way as a virtual node but using resources of a physical link. This being said, a virtual network does not use all the resources of a physical infrastructure, and so it is possible to have several virtual networks over a physical infrastructure. In order to achieve this, it must allocate resources from physical nodes and physical links to each virtual network. Information about resource allocation algorithms and technology can be found at [24] and [25].

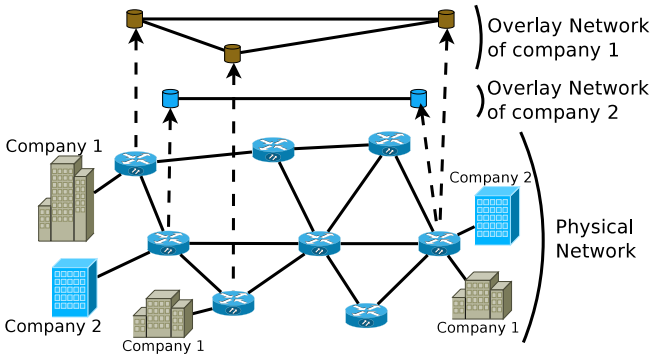


Figure 2: Example of overlay networks

Figure 2 shows an example of a physical infrastructure hosting two overlay networks. We can see that one physical node belongs to both overlay networks as it hosts one virtual node from each overlay network and that some links must transit data from both overlay networks. Each overlay network must be isolated when sharing the same infrastructure. Tunneling is therefore mandatory to keep data from exiting a virtual network.

3.2 Tunneling

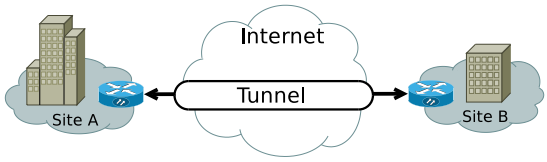


Figure 3: Tunnel

Figure 3 shows the concept of tunneling the data through a tunnel from one point to another. Data using this tunnel is isolated from the rest of the network.

In [26] Cisco defines Tunneling as "a technique that enables remote access users to connect to a variety of network resources (Corporate Home Gateways or an Internet Service Provider) through a public data network." This definition is represented in Figure 3 as we have two sites (A and B) interconnected through the Internet.

In [27], "A tunneling protocol is one that encloses in its datagram another complete data packet that uses a different communications protocol. They essentially create a tunnel between two points on a network that can securely transmit any kind of data between them." Additionally, in [28], "Tunneling enables the encapsulation of a packet from one type of protocol within the datagram of a different protocol." These two definitions add the notion of a packet being encapsulated in another packet, thus the tunneling protocol creates a new packet from the original packet. For example GRE adds a new header (Figure 5) to the packet.

3.3 Multitenancy via virtual networks and tunneling

As stated in Section 3.1 a virtual network only uses a portion of the physical infrastructure's resources. This implies that the rest of the resources can be used for other virtual networks in order to maximize infrastructure usage. Doing so means that the physical resources are shared among virtual networks, however a virtual network belongs to a tenant, and thus the infrastructure provides resources for multiple tenants. This is what we call multitenancy (Section 2).

To grasp the notion of isolation in a data center, it is necessary to understand that the goal of the data center operator is to maximize the use of its infrastructure. To achieve this, it uses virtual networks and tunneling in order to accommodate the maximum number of tenants possible on its infrastructure. Several hundreds or thousands tenants can share the same infrastructure

inside a data center, thus the main challenge when providing isolation in this environment is to be able to provide it for a very large number of tenants. Each tenant wants to have its network isolated from other tenants, therefore scalability is a concern. Another challenge is to provide an isolation solution that can sustain misbehavior or misconfiguration inside tenants' networks without impacting other tenants. Therefore the solution must be resilient. Additionally, isolation inside a data center must be assured inside the whole data center therefore all the devices composing the infrastructure must manage the chosen isolation solution. A fourth challenge is to maintain availability of the data center even when updating the infrastructure by adding new devices or tenants. Moreover, each tenant has its own rules and policies, thus the isolation solution must enforce those rules only for the right tenant.

To summarize, the main issue with multitenancy in a data center is caused by the huge numbers of tenants, policies, servers and links. It is mostly a scalability issue. However, the isolation solution must cope with this issue without degrading, too much, the performances of the infrastructure. Therefore, another challenge for multitenancy is to have an isolation solution with a low overhead.

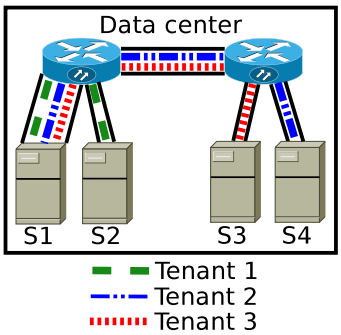


Figure 4: Multitenancy

An example of a multi-tenant data center is shown in Figure 4. In this example, either the three tenants have data stored on Server 1 (S1) and each tenant possesses a virtual network therefore their traffics are tunneled from end to end. The various flow representations indicate all path long isolation for each tenant's traffic while using a common infrastructure.

In order to enforce multitenancy we need both isolation in the network (Section 3.3.3), via the tunneling protocol, and isolation in the nodes, via Hypervisor level Isolation (Section 3.3.1) or Database level Isolation (Section 3.3.2). However, by achieving isolation there is some performance degradation (Section 3.4.1) as well as risks (Section 3.4.2) when the isolation solution is violated.

3.3.1 Hypervisor level isolation

A hypervisor is a software mapping of the physical machine commands to a virtualized machine (VM) running a regular OS. The hypervisor intercepts the OS system calls of the VM and maps these calls to the underlying hardware. This implies that the hypervisor induces a certain percentage of overhead. However, a hypervisor allows the partition of the hardware, thus having several tenants on the same physical node. Since the hypervisor is between the VM and the node, it intercepts all the traffic, thus it can isolate each virtual machine and their resources. With this isolation, a hypervisor allows the protection of tenant resources thereby enabling multitenancy.

3.3.2 Database level isolation

While hypervisor-level isolation is used in Infrastructure-as-a-Service (IaaS), database-level isolation is used in Software-as-a-Service (SaaS). In SaaS, tenants share a database. In [29] the authors describe the three main approaches for managing multi-tenant data in database. The first approach is to have separate databases for each clients. The second one is to have a shared database but with separate schemas, therefore multiple tenants use the same database but each tenant possesses its own set of tables. The last approach is to share both the database and the schemas. In this case an id is append to each record in order to indicate which tenant is the owner of the record. In [30] a new schema-mapping technique for multi-tenancy called "Chunk Folding" is proposed in order to improve the performance of database sharing among multiple tenants. Additionally, in [31], the authors propose a solution called SQLVM which focuses on allocating resources for tenant dynamically while ensuring low overheads. Other solutions like [32, 33, 34] focuses on improving database performances when the number of user increases.

This type of isolation has more security concerns than hypervisor-level isolation. If the modification of the request is mis-configured or if there is an error in an access control list then tenants' information is at risk.

3.3.3 Network level isolation

Tunneling is a key element for isolation inside a data center, however data centers have different constraints than typical LAN networks. The most important one is the number of different tenants whose isolation must be provided. Therefore the scalability of the tunneling protocol and the maximum number of tenants it can manage, is a criterion to take into account when choosing a tunneling protocol. If the tunneling protocol can not isolate all the tenants of a data center then there is no interest in using it, therefore we performed a scalability comparison in Section 7.5. Another criterion is the overhead induced by the tunneling protocol. The challenge is to have the lowest overhead possible. In Section 7.2 we compare the overhead induced by the tunneling protocol. A third criterion, which influences overhead, is the security provided by the tunneling protocol (Section 7.1.4). Then the resilience criterion is

also important in order to quickly mitigate any link or node failure (Section 7.4). As we focus on data center networks we also choose as criterion, the ease of multi data center interconnection which we describe in Section 7.6.

Another element of choice when deciding which tunneling protocol to choose is how it enforces its tunnel. There are two possibilities as indicated in [35]. The first one is the Host isolation technique described at the beginning of Section 4.1. In this technique, the ingress and egress nodes are the ones enforcing the isolation. The second technique, called the Core isolation technique (Section 4.2), enforces network isolation at each switch on the path.

However, while these criteria must be taken into account, the goal is to have the greatest scalability possible with the lowest overhead and complexity.

3.4 Impact of isolation

Providing multitenancy is a key function for cloud data center. However, this functionality implies overhead (Section 3.4.1) and also risks (Section 3.4.2) in case the tenants isolation fails.

3.4.1 Isolation performances overhead

Hypervisor performances have already been studied in several papers [36, 37, 38, 39]. They induce overhead thus performance is decreased. However, not all hypervisors have the same impact on performance. In [36] four hypervisors (Hyper-V, KVM, vSphere, Xen) are compared over four criteria (CPU, Memory, Disk, Network). Their results show that hypervisor overhead is globally low, therefore they do not deteriorate performances too much.

Network isolation solutions are the main subject of this paper, thus in Section 7 a comparison is done between several of them.

First we study their complexity based on six criteria. The first is the control plane design of the solution. The second is network restrictions imposed by each solution, some of them only work with Layer 3 (L3) networks, other only with Layer 2 (L2) networks, and some of them need specific architecture. The third criterion focuses on tunnel configuration and establishment. We analyze if there are messages needed to establish a tunnel, or if it must be allocated before hand on each node. The fourth criterion is tunnel management and maintenance in order to determine the quantity of messages needed by each protocol to keep alive those tunnels. The fifth criterion is the capacity of those tunnels to handle multiple protocols. The sixth, and last, criterion of this complexity study focuses on their security mechanisms.

Then we study the overhead induced by each solution, followed by their capability to migrate VM and a comparison of their resilience. The last criterion for their comparison is their scalability and their capacity to be managed among multiple data centers.

3.4.2 Isolation violation risks

Multitenancy is based on sharing the underlying physical infrastructure, thus tenants' data is stored on the same devices while being isolated via tunneling protocols and hypervisor- or database-level isolation. However, if one of these isolation mechanisms fails, then the data can be seen by other tenants. Having the hypervisor- or the database-level isolation fail implies that all the tenants sharing the same hypervisor or database can access all the data managed by the hypervisor or that is inside the database. This is an issue, however it is restricted to one node and can be resolved without shutting down the whole data center. The worst case is when the tunneling protocol fails. In this case all tenants' traffic is visible, thus data can be stolen or misused by other tenants. To resolve this situation, the data center must be stop in order to reconfigure or change the tunneling protocol, thus the choice of a tunneling protocol is an important decision.

4 Network isolation in traditional data center

The network solutions introduce in this Section were designed before the development of cloud data center. They possess capabilities for isolating flows in a network but are either not scalable enough, or were not designed for cloud data center topologies. As such they can not cope with the increasing number of flows and Virtual Machines (VMs) to isolate. Additionally they can not manage VM live migration, which is not necessary for traditional data centers in which there is no VM, but is mandatory for cloud data centers. Therefore, we present those solutions because they can be used in some traditional data center and mostly to show that there is a need for new multi-tenant network isolation solutions.

4.1 Host isolation

Host isolation is an isolation method which selects flows once they arrive at the destination host or the egress node of the tunnel. This means that all along the path no switching or routing device checked the packets or messages of the flow. The switching or routing is done normally using the information of the transport header. It also means that there is no explicit need for a tunneling protocol in this kind of isolation. For example the Ethernet protocol is a Host isolation protocol. When a packet reaches a host, this host checks the MAC address and accept or not the packet if the MAC address of the packet matches the MAC address of the host. It is only once the packet arrives at the destination host or the egress node of the tunnel that checks are done. Either the destination host or the egress node verifies if both the destination Virtual Machine (VM) and the flow, to which the data belongs, are from the same virtual network. If they do belong to the same virtual network then the data is either delivered to the

VM or dropped depending on policies. This Host isolation advantage is that it does not require the node to know the whole topology. Indeed it is not necessary to know the location of the others VMs in order to distribute the flows. However this techniques has drawbacks. The lack of information about the VMs belonging to the same network imposes that the flows of each VM be propagated in the whole data center. This creates useless traffic, overloading the data center, which is dropped when received by a physical node not belonging to the same network. Additionally, such isolation technique security is weak against a "man in the middle" attack. An attacker who is able to put a listening device in the network could see the traffic from all the clients.

4.1.1 Host isolation for both Layer 2 and Layer 3 networks

The protocol presented in this section can be used over both Layer 2 and Layer 3 network.

4.1.1.1 GRE: Generic Routing Encapsulation

The Generic Routing Encapsulation protocol was proposed in "RFC 1701" [40]. In this RFC, the goal of GRE is to encapsulate any other protocol with a simple and lightweight mechanism. To do that GRE adds a header to the packet (Figure 5). The original packet is called the payload packet and the GRE header is added to it. Then, if the packet needs to be forwarded, the delivery protocol header is also added to the packet. The GRE header is composed of nine mandatory fields, for a total of 4 bytes, and of five optional fields with a variable total length. In these optional fields, there is a routing field which contains a list of Source Route Entries (SRE). Thanks to this field, it is possible to use the GRE header to forward the packet without adding a delivery header.

In the second RFC, "RFC 2784" [41] derived from the original "RFC 1701" without superseding it, the GRE header has been simplified. The header is now made of 4 mandatory fields (4 bytes) and of 2 optional fields (4 bytes). The header length is now limited to 8 bytes whereas in the first RFC there was no length limit. The new header needs the delivery header because there is no information to forward or route the packet in it anymore. As it is a lightweight mechanism, some functionalities are not managed such as the discovery of the MTU along the path. This could be an issue if the source sends a packet with the "don't fragment bit" set in the delivery header and the packet is too big for the MTU. In this case the packet is dropped along the path. As the error message is not required to be sent back to the source, the source could keep sending packets too big for the MTU. Those packets would always be dropped and would never reach their destination.

GRE allows for an easy deployment of IP VPN and can tunnel almost any protocol through those VPN. Additionally it is possible to authenticate the encapsulator by checking the Key field. However, the provisioning is not scalable. In addition, GRE does not protect the payload of its packets because of a lack of integrity check and encryption. In order to resolve this last issue, it is possible to use GRE over IPSec.

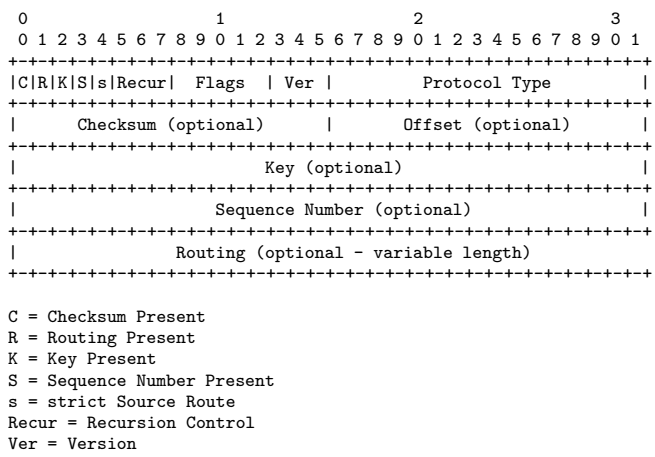


Figure 5: GRE Header

4.1.2 Host isolation for Layer 3 networks

In this section we introduce four Host isolation protocols which impose that the underneath network be a Layer 3 network.

4.1.2.1 PPTP: Point-to-Point tunneling protocol

The Point-to-Point Tunneling Protocol (PPTP), from "RFC 2637" [42], was introduced 5 years after the first version of GRE [40]. This can explain why GRE is used to do the tunneling in PPTP. However, the GRE header is modified in PPTP (Figure 6). The Routing field is replaced by an acknowledgment number field of 4 bytes. This way, the header has a maximal length. The new acknowledgment number field is used to regulate the traffic of a session. As the PPTP tunnel multiplexes sessions from different clients, this acknowledgement number allows traffic policing for each session. The tunnel formed by PPTP between the PPTP Access Concentrator (PAC) and the PPTP Network Server (PNS) is deployed over IP, which is why the routing field was not necessary (as the routing is done by IP). The other difference with GRE is the use of the key field. In PPTP, the key field is divided in two parts: the higher two bytes, which are used for the payload length, and the lower two bytes which are for the call Id. The call Id represents the owner of the packet.

One of the advantages of PPTP is that it only needs two devices to be deployed at each end of the tunnel. A PNS at one end and a PAC at the other. There is no need to know or interact with the network between both ends. For data confidentiality, PPTP uses an encryption technique called Microsoft Point-to-Point Encryption (MPPE). MPPE uses the RSA RC4 encryption algorithm and a session key to encrypt the packet. Three key lengths are supported: 40 bits, 56 bits, and 128 bits. Another advantage of PPTP is that there

is no need for an agreement with the service provider. The administrator in charge of the PPTP session has complete control over it. However the administrator must be able to install, configure, and manage a PPTP device at both ends of the tunnel.

The disadvantage of PPTP is that it uses TCP for all its signaling so it does not support multipath, and always uses the same path for a session. PPTP is end-user initiated because of its design. Only both ends of the tunnel know about the tunnel, so the service provider is not aware of PPTP. Because of that there is no Quality of Service (QoS) possible.

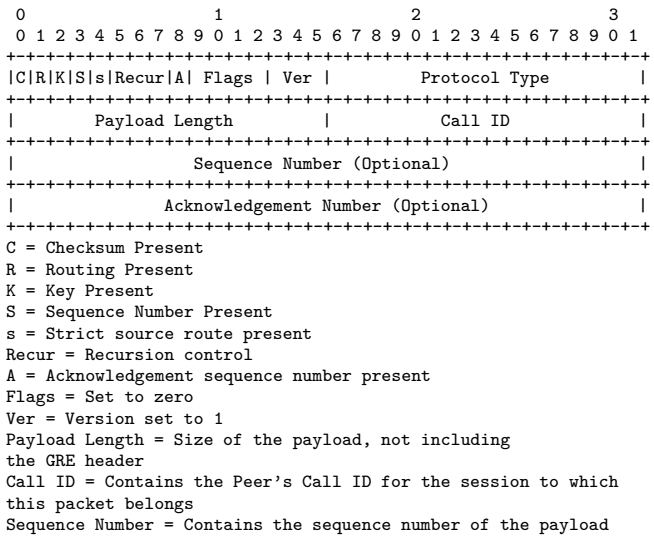


Figure 6: PPTP Header

4.1.2.2 L2TP: Layer Two Tunneling Protocol

L2TP was developed by the IETF and proposed as a standard in "RFC 2661" [43]. L2TP is designed to transport PPP frames (Layer 2) over a packet-switched network (Layer 3). With L2TP it is possible to have two PPP endpoints residing on different networks interconnected by a Layer 3 network. It allows extending the Layer 2 network to other Layer 2 networks interconnected through a Layer 3 network. To design L2TP, the IETF used the Layer-2 Forwarding (L2F) and the Point-to-Point Tunneling Protocol (PPTP) as a starting point. L2F is a Cisco proprietary tunneling protocol which provides a tunneling service for PPP frames. PPTP was developed by Microsoft and is also designed to transport PPP frames over Layer 3 networks. L2TP works with two devices, the L2TP Access Concentrator (LAC) and the L2TP Network Server (LNS). Those are the endpoints of the L2TP tunnel. The LAC is located at the ISP's Point of Presence (POP). The LAC exchanges PPP messages with users, and communicates with customers' LNS to establish tunnels. To use L2TP, the ISP needs to be informed because they must have a L2TP-capable POP. This POP must encapsulate PPP frames within L2TP ones, and forward them through the correct tunnel toward the LNS, which belongs to the customer. The LNS must accept L2TP frames, and strip the L2TP encapsulation in order to deliver the PPP frames to the appropriate interface. L2TP possesses integrated security techniques such as an authentication between the client and the LAC at the initiation of the tunnel, a tunnel authentication between the LNS and the LAC, and a client authentication and authorization between the client and the LNS. This last authentication can use the Password Authentication Protocol (PAP), the Challenge Handshake Authentication Protocol (CHAP), or a one time password. After that, the PPP session begins and the data can be exchanged.

With L2TP, the ISP is needed which results in an extra cost in order to establish the tunnel. Nevertheless with the ISP's involvement, it is possible to add Quality of Service guarantees (QoS) and to benefit from the ISP IP network reliability. The fact that the L2TP encapsulation is done by the LAC means that there is no need for client software. This is advantageous because it removes the difficulties associated with managing remote devices. However there is no multipath management because of the design of L2TP, in which a client is in one tunnel and the tunnel has only one path. But load balancing and redundancy are possible thanks to multiple home gateways.

4.1.2.3 L2TPv3: Layer Two Tunneling Protocol - Version 3

In L2TPv3, "RFC 3931" [44], the tunnel can be established between L2TP Control Connection Endpoint (LCCE) which are the LAC and the LNS. The novelty is that it is possible to have a LAC-to-LAC tunnel or a LNS-to-LNS tunnel. In addition a device can be a LAC for some sessions and a LNS for others. Another modification is the use of two headers, the control message header (Figure 7) and the data message header (Figure 8), instead of one header for all messages.

The control message header has the same length as the original one but with one less field. The Session ID field is now 4 bytes long, instead of 2 bytes, and the Tunnel ID field is removed. L2TPv3 replaces the data header with a L2TPv3 Session Header. The RFC states that, "The L2TP Session Header is specific to the encapsulating Packet-Switched Network (PSN) over which the L2TP traffic is delivered. The Session Header MUST provide (1) a method of distinguishing traffic among multiple L2TP data sessions and (2) a method of distinguishing data messages from control messages."

The LCCE from L2TPv3 does not need to be at a Point Of Presence (POP) of an ISP. Consequently, it is possible to establish a tunnel without the ISP's help, thus reducing the cost. However, without the ISP there is no service guarantee on the Layer 3 network.

4.1.2.4 PWE3: Pseudo Wire Emulation Edge-to-Edge

Pseudo Wire Emulation Edge-to-Edge (PWE3) is a technology that emulates services from Layer 2 such as Frame Relay, ATM, Ethernet over packet switched networks (PSN) using IP or MPLS. It was proposed in

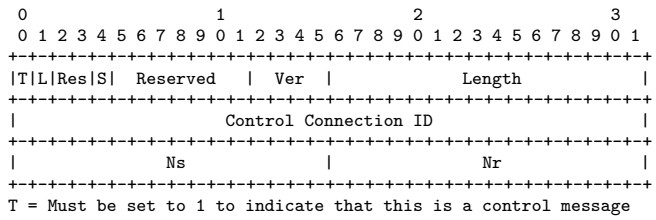


Figure 7: L2TPv3 control message header

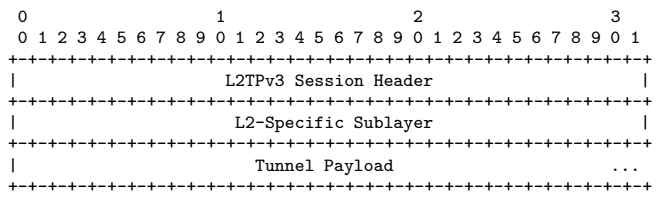


Figure 8: L2TPv3 data message header over IP

"RFC 3985" [45]. PWE3 defines an encapsulation function which encapsulates service-specific bit streams, cells, or PDUs. This service-specific data is encapsulated at the ingress node in order to be sent over the PSN. The encapsulation is done by the Provider Edge (PE), then the data is carried across a PSN tunnel. PWE3 is an encapsulation protocol which emulates a Layer 2 service over a PSN network. However, to tunnel the data, it needs a tunneling protocol such as L2TP or MPLS. This protocol uses a Host isolation method if it is used with L2TP and a Core isolation method if it is used with MPLS.

4.2 Core isolation

The Core isolation method requires that each node of the network possesses a wider knowledge of the topology than when using the Host isolation method. As a matter of fact, in the Core isolation method each node on the path (switch or router) has to check the packet in order to verify if it can be forwarded toward its destination. The benefit of such an isolation method is that the packet is dropped at the closest node from the source if the destination is not reachable for policy reasons. This method considerably reduces traffic, by preventing the transmission of useless traffic to nodes which are not concerned by such traffic. However it is necessary to have a global view of the network topology in order to transmit packets. This implies either a pre-configured network with strict rules or an auto-configurable network. The first case means that the topology is rigid which is contrary to virtualization principles such as live migrations. In the second case, it would increase the waiting time before being able to make two entities of the network communicate. This increase happens due to the time needed for exploring and sharing the network's information.

4.2.1 Core isolation for Layer 2 networks

Both Core isolation protocols introduce in this section require the underneath network to be a Layer 2 network.

4.2.1.1 VLAN: Virtual LAN

A VLAN emulates an ordinary LAN over different networks as defined in the "802.1Q IEEE standard" [46]. The nodes belonging to a VLAN are members of this VLAN. A member of a VLAN communicates with the other members of the VLAN as if they were on the same LAN despite their geographical location. VLAN members are in a logically separated LAN and share a single broadcast domain. They do not know that there are not on the same physical LAN. The other nodes, not member of the VLAN, will not see the traffic from the VLAN and will not receive any of the broadcast messages from the VLAN. All the traffic from a VLAN is isolated from the rest of the network.

There are three methods to recognize the members of a VLAN. The first method is port based. The switch knows that the node connected at the specified port is a VLAN member. The specified port is tagged and is now processing VLAN-only messages. The second method is based on the recognition of the MAC address. And the third method is based on the recognition of the IP address. Independent of the method, the packets of the VLAN are tagged with a 4-byte header (Figure 9) between switches and routers. This field contains the VLAN ID (VID) field, which is 12 bits long. The VID is used to know which VLAN the message belongs to, since switches and routers can multiplex VLANs on a link. Such link is called a VLAN trunk.

If a VLAN member has moved and the VLAN is configured to use MAC addresses, the VLAN can recognize that the member has moved. The VLAN can then automatically reconfigure itself without the need to change the member's IP address.

Among VLAN advantages we have that VLANs facilitate administration of logical groups of stations. They allow stations to communicate as if they were on the same LAN. The traffic of a VLAN is only sent to members of the VLAN which allows flow separation. A VLAN diminishes the size of a broadcast domain and so improves bandwidth. There is also a security improvement thanks to a logical isolation of the VLAN members. An ISP agreement is not needed to establish a VLAN.

A disadvantage of VLANs is that there are only 4096 VLANs because of the size of the tag. To solve this, the IEEE 802.1ad standard [5], presented in Section 5.2.1.1, has been developed and it increases the number of VLANs. Another issue in the original definition of the 802.1Q is the lack of a control plane which enable

automatically provisioning the path on each switch. This last issue is fixed by the use of the GARP VLAN Registration Protocol (GVRP) [47], which is a Generic Attribute Registration Protocol application.

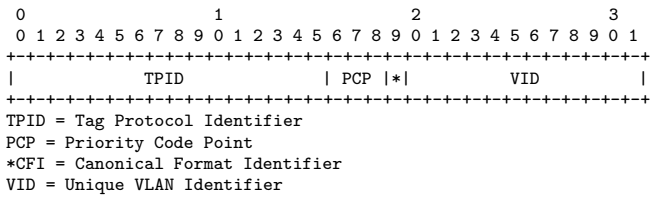


Figure 9: VLAN Header

4.2.1.2 802.1ad - Provider Bridges

The first draft [48] of the 802.1ad standard was released in 2002 and was intended to enable a service provider to offer separate LAN segments to its users over its own network. Therefore, both the user and the provider possess their own VLAN field which increase the number of available VLANs. Even if this solution was proposed before the growth of cloud data center, it has been adapted to fit to cloud data center networks, therefore we present this solution in Section 5.2.1.1.

4.2.2 Core isolation protocols for Layer 3 networks

In this section we present three Core isolation protocols which can be used only if the underneath network is a Layer 3 network.

4.2.2.1 MPLS: Multiprotocol Label Switching

Multiprotocol Label Switching (MPLS), defined in "RFC 3031" [49], is a circuit technique that uses label stacks on packets in order to forward them. MPLS uses Layer 3 (IP) routing technique with Layer 2 forwarding in order to increase the performance/price ratio of routing devices and to be open to new routing services invisible at the label forwarding level.

MPLS decreases the processing time of each packet, with only a label of 20 bits (Figure 10) to look at to forward the packet. It has the ability to work over any Layer 2 technology such as ATM, Frame Relay, Ethernet, or PPP. MPLS has traffic engineering techniques with the Resource reSerVation Protocol (RSVP) [50] or Constraint-based Routing Label Distribution Protocol (CR-LDP) [51] and enables Quality of Service (QoS) using DiffServ [52].

MPLS packets are named "labeled packets" and the routers which support MPLS are called "Label Switching Routers" (LSR). The packets are labeled, at the ingress LSR, depending on the forwarding equivalence class (FEC) they belong to. Those labels are locally used, each LSR changes it depending on the label the next LSR in the path as announced for the FEC. For each FEC exists at least one path across the MPLS network. This path is a Label Switched Path (LSP). All the packets of one FEC takes the same LSP. On this LSP, the labeled packets are forwarded based on their labels. After all the label changes, the egress LSR removes the label.

MPLS creates a tunnel for the traffic from the rules it uses. It is also possible to manually edit the labels to define a LSP through the MPLS network. The traffic is tunneled all along the path thanks to the configuration and rules established in the control plane.

In order to use MPLS, the network must be MPLS-ready and configured with a FEC for the traffic. ISP intervention is needed to have a FEC configured, meaning extra cost for the client. However customers could have QoS for their traffic.

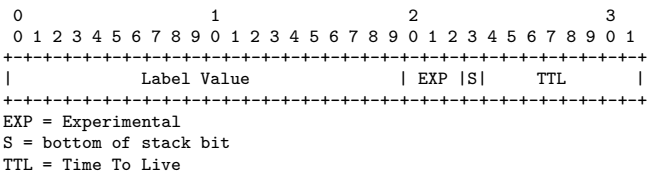


Figure 10: MPLS header

4.2.2.2 GMPLS: Generalized Multi-Protocol Label Switching

GMPLS was proposed in "RFC 3471" [53] and updated in "RFC 3945" [54] as an extension of MPLS. This extension adds support for new switching types such as Time-Division Multiplexing (TDM), lambda, and fiber port switching. To support those new switching types, GMPLS has new functionalities which modify the exchange of labels and the Label switched Path (LSP) unidirectional characteristic. MPLS forwards data based on a label, but the new switching techniques are not based on header processing, so GMPLS must define five new interfaces on the Label Switching Routers (LSR).

1. The Packet Switch Capable (PSC) interface, like the one from MPLS, uses the header of the packet for routing.

2. The Layer-2 Switch Capable (L2SC) interface uses the frame header, like the MAC header or the ATM header, to forward the frame.

3. The Time-Division Multiplex Capable (TDM) interface switches data thanks to the data's time slot in a repeating cycle.

4. The Lambda Switch Capable (LSC) interface receives data and switches it via its wavelength when it was received.

5. The Fiber-Switch Capable (FSC) interface switches the data based on its position in physical space.

For the LSC interface, the header is 32 bits long and contains only a Label field (Figure 11). The other interfaces use the same header which contains a Label field with a variable length (Figure 12). However, to establish a circuit, two interfaces of the same type are

needed at each end. In GMPLS it is possible to establish a hierarchy of LSPs on the same interface or between different interfaces. If it is on the same interface, it means that the LSR was able to multiplex the LSPs. If it occurs between interfaces then that means that the LSP start with one type of interface and another one is used along the path. If such an interface change happens on the path, then the original LSP is nested into another LSP. This new LSP must end before the original LSP in order to have the same interface type for the final one as the first one. For example, the LSP starts and ends on a PSC interface and along the way the interface changes into FSC so the PSC LSP is nested into a FSC LSP. As MPLS, GMPLS uses the control plane to establish rules, labels, to route all the data of an LSP which are tunneled through a unique path. GMPLS extends LSRs capabilities from MPLS by allowing different techniques for data forwarding. However GMPLS shares the same constraint as MPLS, in that there must be an agreement with the ISP before using it.

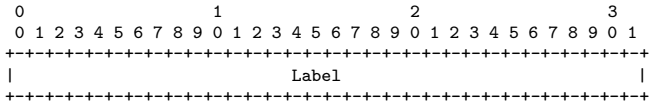


Figure 11: GMPLS header for Lambda interface

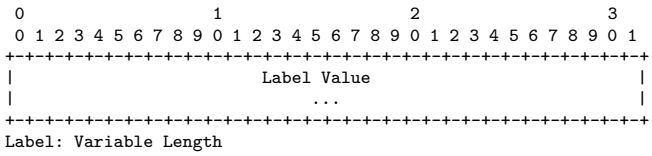


Figure 12: GMPLS header for PSC, L2SC, TDM and FSC interfaces

4.2.2.3 BGP/MPLS IP Virtual Private Networks

In Border Gateway Protocol (BGP) / MultiProtocol Label Switching (MPLS), described in "RFC 4364" [55], the idea is to use the MPLS label to forward the packet in the network and BGP to exchange the route information between the LSRs. As shown in Figure 13, clients need to install at least one Customer Edge (CE) router at each site they want to connect. The CE has to know every private IP address from the site it is in. The CEs are then connected to Provider Edges (PE) provided by the ISP. Each PE learns all the IP addresses accessible through the CE it is connected to and then uses BGP to exchange the addresses with the other PEs over the ISP's core network. The PE creates one or more Virtual Routing and Forwarding (VRF) table(s) containing the information of the path to each PE and each device in his local network. The core network router, which is working with MPLS, does not know any of the clients addresses.

A Virtual Private Networks (VPN) contains at least two PEs to connect two sites. In order to create such a network, the client and the ISP must have an agreement. However a BGP/MPLS VPN system allows the overlapping of address spaces between VPNs, so clients could use any addresses they want in their VPN. BGP/MPLS IP VPNs grants privacy if the network is well configured. But there is no encryption, no authentication, and no integrity check method. In order to add security measures IPsec must be used.

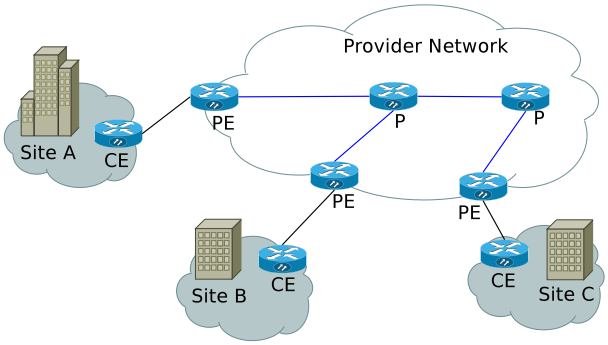


Figure 13: BGP/MPLS network

5 Network isolation in cloud data center with multi-tenant capabilities

The solutions and protocols shown in Section 4 are not appropriate for isolation in Cloud Data Centers (Cloud DC) prone to virtualization. New data centers use virtualization technologies in order to increase the number of tenants sharing the same infrastructures and the limits of those isolation techniques are not sufficient for accommodating all clients.

For example, GRE 4.1.1.1 provisioning is not scalable. We must know beforehand how many clients there will be in the data center, which is not possible in virtualized data centers. PPTP is end user-initiated because of its design, thus preventing the administrator of the data center from using it freely.

The VLAN limit of 4096 different identifiers is very small in comparison to the number of clients in a virtualized data center. Already in 2010, VMware users were running an average of 12.5 virtual machines per physical server [56]. However those users were not necessarily professional data center, and consequently may not have purchased the best server. In [57] the number of VMs per server is given as a ratio of 25 VMs per 1 server, and is expected to grow to 35 VMs per server. However, VMware is currently advertising that some of its servers can host 29 VMs [58]. We need to be careful with these statements because we do not know what kind of VMs, work-intensive or not, they take into account in their calculations. Nevertheless, sticking with 29 VMs per server and each VM belonging to a different client, the VLAN limit is reached with 141 servers, which is a small number of servers for a cloud data center. In [59] the number of servers expected in Amazon data center is 46.000 and in [60] we have

approximations of the number of servers owned by the largest data center companies.

MPLS (Paragraph 4.2.2.1) could be a solution for Layer 3 data centers as it has enough different labels to accommodate for more than a million customers. However it is not widely used in data centers because of a complexity issue concerning the distribution of the labels over the network and because of its cost [61].

In this section we group solutions based on two criteria. The type of isolation (Host isolation or Core isolation) is the first criterion. Then, the second criterion to further separate the solutions is the layer of the underneath infrastructure required by the solution.

5.1 Host isolation for Cloud DC

In this section we present 8 solutions which we consider as Host isolation solutions designed for Cloud DC. Those solutions are: Diverter, BlueShield, NetLord, LISP, NVGRE, STT, VL2, and DOVE.

5.1.1 Host isolation for Layer 2 Cloud DC

Protocols introduced in this section use the Host isolation technique and work over a Layer 2 network.

5.1.1.1 Diverter

Diverter [8] creates an overlay network with a software-only approach, thus alleviating the need for manual configuration of switches and routers. The software module, called VNET, is installed on the host of each physical server. The server's packets (VMs and host packets) and the packets from the network are intercepted by the VNET which processes them. During this process, the VNET replaces the MAC addresses of the packet in order to have no virtual addresses appearing on the core network. The destination MAC address is replaced by the MAC address of the physical server hosting the destination VM. The source MAC address is replaced by the MAC address of the server which hosts the source VM. In the Layer 2 core network, the switches perform packet forwarding using the server's MAC address. Tenant isolation is done thanks to the VNET's control of the packet. If there is no rule in both VNETs allowing the communication between the VMs then the packet is not sent by the ingress VNET. If it is mistakenly sent, then the packet is dropped by the receiver VNET. The control of the packet is done two times at both VNETs. This implies that both VNETs must have the same rules allowing this communication between the two VMs.

In Diverter the tenants cannot choose the addressing scheme they want. They must use IP addresses that follow a specific format which is:

10.tenant.subnet.vm

Where *tenant* is the tenant ID, *subnet* the number of the subnet belonging to the tenant in which the VM is present and *vm* is the number of the vm in the subnet. With this addressing scheme there is no risk of having identical addresses.

In conclusion Diverter provides Layer 3 network virtualization, over a large flat Layer 2 network, in which tenants are isolated. Tenants can also control their own IP subnet and VMs addresses as long as they respect the restrictions on IP addresses imposed by Diverter.

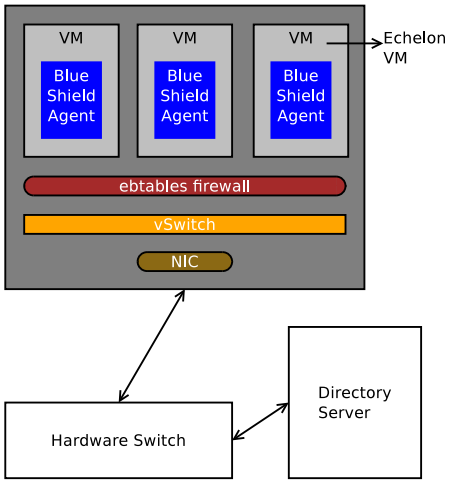


Figure 14: BlueShield architecture (adapted from figure of BlueShied paper [11])

5.1.1.2 BlueShield

BlueShield [11] is an architecture which neither adds a header nor modifies the already existing header. It also does not use a tag or VLAN like value to separate tenants' traffic. Instead, it prevents the tenants' traffic from being sent on the Layer 2 network by blocking address resolution and preventing the configuration of static hardware address entries. With these characteristics, BlueShield provides a complete isolation between tenants.

In order to allow communication between VMs of the same tenant - or not, depending of the tenants' demands - BlueShield uses a BlueShield Agent (BSA) in each VM and a vSwitch at each server (Figure 14). The BSA will see all the ARP requests made by the VM and will convert them in directory look-up (DLU) requests addressed to one or multiple Directory Servers (DSs). The ARP requests are then dropped by the vSwitch of the server before reaching the NIC and the network. The DS searches in its rules whether or not the source VM can communicate with the destination VM. If communication is not allowed, then the DS does not answer and the VMs can not communicate. Otherwise the DS answers the request. As the BSA can send a request to multiple DS, they all answer, so there is a requirement for synchronization of DSs' rules.

BlueShield defines Echelon VMs as those whose task is to increase security and isolation. These Echelon VMs are disseminated on the network and share security rules that they enforce by scanning all the packets passing through them. In order for the traffic to pass through an Echelon VM, the rules in the DS must be modified accordingly. The DS, instead of answering with the MAC address of the destination VM, will send

the Echelon VM MAC address.

In conclusion, BlueShield is a technique which allows tenants' data isolation but not the isolation of tenants' address-space. In addition, the rules in the DS, enforcing this isolation, must be the same on all the DSs and the Echelon VMs, if these last devices are used. The establishment and configuration of these rules lies with the administrator, and the techniques are those of his or her choosing.

5.1.2 Host isolation for both Layer 2 and Layer 3 Cloud DC

In this section we present one Host isolation protocol which can be used over either a Layer 2 or a Layer 3 network.

5.1.2.1 NetLord

In "NetLord: A Scalable Multi-Tenant Network Architecture for Virtualized Datacenters"[13] the authors proposed a new multi-tenant network architecture. The core network is a Layer 2 (Ethernet) network and the use of Layer 3 (IP) is done at the last hop between the edge switch and the server.

To provide tenant traffic isolation in the network, NetLord encapsulates tenant data with both Layer 3 (IP) and Layer 2 (Ethernet) headers (Figure 15). To do so, NetLord uses an agent in the hypervisor of the server to control all the VMs on the server. This agent has to encapsulate, route, decapsulate and deliver the tenant's packet to the recipient.

The source NetLord Agent (NLA) encapsulates the tenant's packet with an IP header and an Ethernet header. The Ethernet destination address of the added Ethernet header is the MAC address of the egress edge switch. The IP destination address of the added IP header is composed of two values. The first is the number of the switch port (P) to which the machine is connected, and the second is the Tenant_ID (TID). This IP address is analyzed at the egress edge switch where it is used to route the packet toward the correct server by using the port number P from the IP address. Then, when the packet is received by the server, it is handed off to the destination NLA. This NLA has to use the TID part of the IP address to send the data to the correct tenant. The use of IP routing at the last hop allows the use of a single edge switch MAC address in order to communicate with the VMs on the server beyond this edge switch. This way, physical and virtual machines, from other servers, will only have one mac address to store, the edge switch MAC address. In addition, the mac addresses of the VMs are not exposed on the core network. However the tenant's ID is exposed in the outer IP header. This exposition can be used by the provider to apply per-tenant traffic management in the core network without the need of per-flow Access Control Lists (ACLs) in the switches.

NetLord also provides address-space isolation for tenants. The tenants are able to use any Layer 2 or Layer 3 addresses because NetLord does not impose restrictions on addresses, and there is no risk of badly routed packets because of these addresses. As stated earlier, the ingress switch will use the MAC address of the egress switch on the core network to forward the data. Then the IP address, composed of the port number and the TID, will be used at the egress switch. At any given time, the addresses defined by the tenant are only visible in the tenant virtual network.

The tenant data between the egress and ingress switches are conveyed over the Layer 2 network thanks to VLANs. In order to choose which VLAN to use, NetLord applies the SPAIN [62] selection algorithm. However to support the SPAIN multipath technique and stock per-tenant configuration information, NetLord uses Configuration Repository which are databases. It also uses the same mechanisms as Diverter [8] to support virtual routing.

To establish a NetLord architecture, edge switches that support IP forwarding must be used, which is not a common feature for commodity switches. In addition, the use of SPAIN implies a scalability issue and there is no support for bandwidth guarantee. In conclusion, NetLord provides tenant isolation but has some drawbacks in other areas.

5.1.3 Host isolation for Layer 3 Cloud DC

The protocols introduced in this section use the Host isolation technique and require a Layer 3 network.

5.1.3.1 LISP: The Locator/Identifier Separation Protocol

The Locator/Identifier Separation Protocol (LISP), presented in "RFC 6830" [2], aims at splitting the routing and the addressing functionalities. Currently, the IP address, a single field, is used both for routing and for addressing a device. In LISP, the routing functionality is done by Routing Locators (RLOCs) and the addressing functionality is done by Endpoint Identifiers (EIDs). An RLOC is an address, the same size as an IP address, of an Egress Tunnel Router (ETR). This RLOC indicates the location of the device in the network. This value is the one used by the Ingress Tunnel Router (ITR) to route the packet through the network toward the ETR. The ETR is the gateway of the private network. Then, to route the packet to the correct node in the private network the EID value is used. All the EID of the private network is mapped in the ETR. This value also has the same length as an IP address (32-bit for IPv4, or 128-bit for IPv6). Such a split is done by using different numbering spaces for EIDs and RLOCs. By doing this, LISP improves the scalability of the routing system thanks to the possibility of a greater aggregation of RLOCs than IP addresses. However in order to have this better aggregation, the RLOCs must be allocated in a way that is congruent with the network's topology. On the other hand, the EIDs identify nodes in the boundaries of the private network and are assigned independently from the network topology.

The encapsulation of the packet, in an IPv4 network, is shown in Figure 16. The outer header is the IP

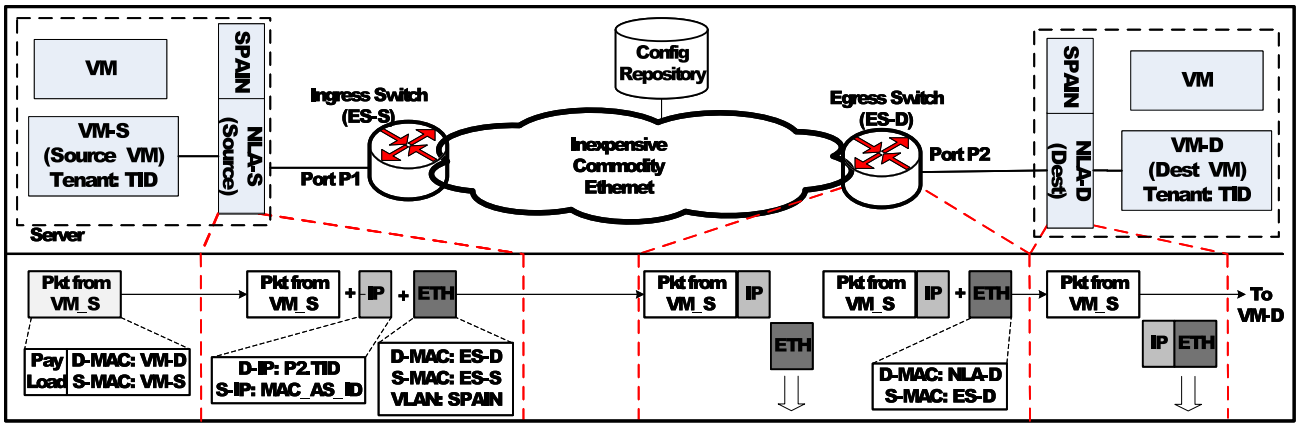


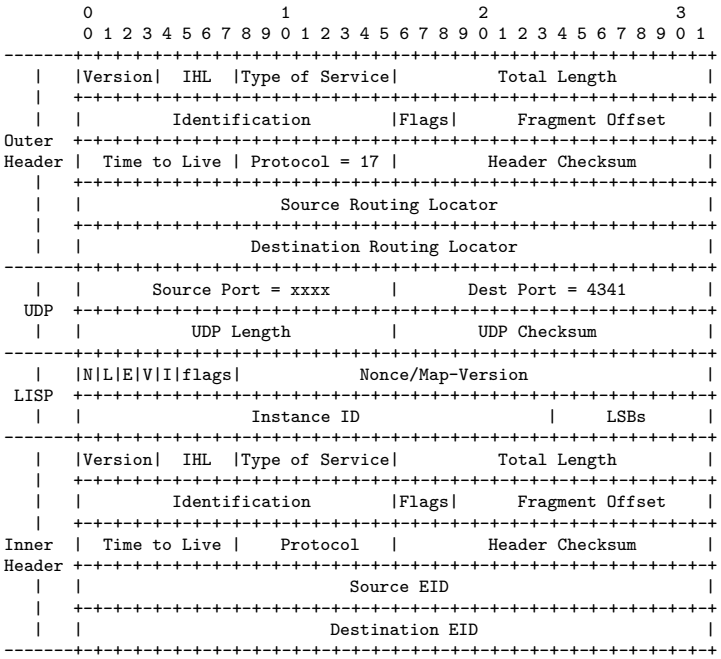
Figure 15: NetLord architecture (figure from NetLord paper [13])

header with the RLOCs addresses as the source and destination addresses. Then the UDP header is added and followed by the LISP header which is 4 bytes long. The inner header is also an IP header but the source and destination addresses are now the EIDs addresses.

It is a tunneling protocol that simplifies routing operations such as multi-homed routing and facilitates scalable any-to-any WAN connectivity. It also improves the scalability of the routing system through greater aggregation of RLOCs. However to benefit from LISP advantages it must use a LISP-enabled ISP. The ISP also benefits from using LISP because it has less information in his routing devices thanks to RLOCs aggregation.

5.1.3.2 NVGRE: Network Virtualization using Generic Routing Encapsulation

Network Virtualization using Generic Routing Encapsulation (NVGRE), detailed in "NVGRE: Network Virtualization using Generic Routing Encapsulation" [3], is based on the Generic Routing Encapsulation (GRE) [40] encapsulation method. NVGRE allows the creation of virtual Layer 2 topologies on top of a physical Layer 3 network. The goal of NVGRE is to improve the handling of multitenancy in data centers. Network Virtualization is used in order to provide both isolation and concurrency between virtual networks on the same physical network infrastructure. To improve isolation, NVGRE modifies the GRE header by replacing the Key field with two fields, virtual Subnet ID (VSID) and FlowID (Figure 17). The first 24 bits are for the VSID field and the following 8 bits for the FlowID field. The VSID is used to identify the virtual Layer-2 network. With its 24 bits it is possible to have 2^{24} virtual layer-2 networks which is more than the 4096 VLANs. The flowID is used to provide per-flow entropy in the same VSID. This NVGRE packet can then be encapsulated in both versions of IP whereas NVGRE cannot contain a 802.1Q tag. The NVGRE tunnel needs NVGRE endpoints between the virtual and physical networks. Those endpoints could be servers, network devices, or part of a hypervisor. NVGRE is using the IP address scalability to lower the size of Top of Rack




```
Inner Header = header on the datagram received from the originating host
Outer Header = header prepended by an ITR
IHL = IP-Header-Length
N = nonce-present
L = 'Locator-Status-Bits' field enabled
E = echo-nonce-request
V = Map-Version present
I = Instance ID
flags = 3-bit field reserved for future flag use
LISP Nonce = 24-bit value that is randomly generated by an ITR when
the N-bit is set to 1
LISP Locator-Status-Bits (LSBs) = set by an ITR to indicate to an ETR
the up/down status of the Locators in the source site when the L-bit
is also set
```

Figure 16: LISP IPv4-in-IPv4 Header Format

switches' MAC address table. For the moment, the fact that NVGRE is a work in progress and not a standard prevents it from being widely deployed, while awaiting possible modifications.

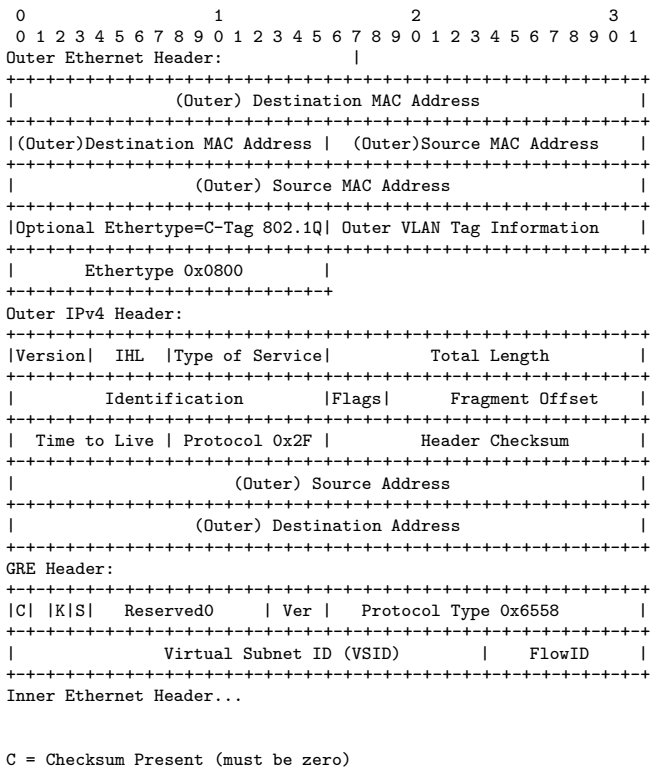


Figure 17: NVGRE Header

5.1.3.3 STT: Stateless Transport Tunneling Protocol

The Stateless Transport Tunneling Protocol (STT), introduced in "A Stateless Transport Tunneling Protocol for Network Virtualization (STT)" [4], is a new IP-based encapsulation and tunneling protocol which adds a new header (Figure 18) to the packet and also modifies the TCP header. The new header contains a 64-bit Context ID field which can be used to differentiate $2^{64} \approx 1.8 \times 10^{19}$ virtual networks. The modifications done to the TCP header are about the meaning and use of both the Sequence Number (SEQ) and the Acknowledgment Number (ACK). The SEQ field is now divided into two parts. The upper 16 bits of the SEQ field are used to indicate the length of the STT frame in bytes. The second part of the SEQ field, the lower 16 bits, is used for the offset, expressed in bytes, of the fragment within the STT frame. Reusing the TCP header allows the STT to be easily encapsulated in IP datagrams. The Protocol Number field for IPv4 or the Next Header field for IPv6 have the same value as for regular TCP. An additional difference between TCP and STT is that STT, as the name indicates, does not use state.

The goal of STT is to tunnel packets efficiently so it supports Standard Equal Cost Multipath (ECMP). Nevertheless, STT imposes that all the packets belonging to the same flow follow the same path. The multipath is done on a flow basis and not a packet basis. However, the most important drawback of STT is the fact that there must not be any middle boxes on the path. If those middle boxes are present, then they have to be configured to let STT frames pass through. This implies that access to the middle boxes is required. So, for the moment, it is not feasible to have an STT tunnel between two sites linked by an unmanageable network. In addition, STT is not a standard, so not all devices will be able to work with it.

Figure 18: STT header

5.1.3.4 VL2

VL2, presented in [15], is an architecture designed to allow agility, and notably the capacity to assign

any server to any service. To do so, VL2 uses two different IP address families, the Location-specific IP addresses (LAs) and application-specific IP addresses (AAs). This addressing scheme separates server names, the AAs addresses, and their locations, the LAs addresses. However, this implies that a mapping between AAs addresses and LAs addresses is needed. This mapping is created when application servers are provisioned to a service and assigned AAs addresses. This mapping is then stored in a directory system which must be reliable. To improve this reliability the directory system can be replicated and can use several directory servers but this implies that those directory servers are synchronized. The directory system is used to achieve addresses resolution but every server must implement a module called a VL2 agent to contact the directory system. This VL2 agent contacts the directory system to retrieve the LA address corresponding to an AA address. Each AA address is associated with an LA address. This LA address is the identifier of the Top of the Rack switch to which the server, identified by the AA address, is connected. The AA address remains the same even if the LA address is changed due to a virtual machine migration or re-provisioning. The AAs addresses are assigned to servers and the LAs addresses to the switches and interfaces. To do this LA address assignment, switches run an IP-based link state routing protocol.

VL2 works over a Clos topology [63] and a Layer 3 network. This network routes traffic by LAs addresses, so in order to route the traffic between servers with AA addresses, encapsulation is needed. This encapsulation is done by the VL2 agent which encapsulates the IP packet in an IP packet (IP-in-IP) and uses the associated LA address, in the directory system or its local cache, with the AA address destination address.

In VL2, the isolation of the server is achieved through the use of rules in the directory system. Additionally, those rules are enforced by each VL2 agent. For example if a server is not allowed to send a packet to a different server, the directory service will not provide an LA address to the VL2 agent for the packet which will be dropped by the VL2 agent.

5.1.3.5 DOVE: Distributed Overlay Virtual nEtwork

Distributed Overlay Virtual nEtwork [16, 17] is a technique designed with a centralized control plane over a Layer 3 network. DOVE does not provide an encapsulation protocol and uses others protocols, such as VXLAN, NVGRE, or STT, as long as those protocols allow the use of two parameters.

The first parameter is the virtual network ID and the second, a policy specifier defined by a domain ID which is optional. By not providing an encapsulation protocol, DOVE is not limited to Ethernet emulation and could be used over a Layer 2 network.

Dove provides tenant isolation thanks to the use of an encapsulation protocol whose header is added by dSwitches, and the use of the DOVE Policy Service (DPS). The dSwitches are the edge switches of the DOVE overlay network. They are used in each physical server to act as the tunnel endpoint for the VMs of these servers. The DPS is a unique component in the DOVE network, whose function is to process dSwitches policy requests. It maintains all the information regarding existing virtual networks as well as their correlation with the physical infrastructure, policy actions, and rules. It is thanks to these policy requests and responses that a dSwitch knows if a VM can communicate with a different VM, and learns the address of the dSwitch which manages the other VM.

As a solution using a centralized control plane, DOVE needs a "highly available, resilient, and scalable" device to host the DPS as stated by the authors. As we have seen in the PortLand architecture, the Fabric manager needed at least 15 CPU cores working nonstop to process the ARP requests for 27.648 end hosts which each make 25 ARP requests per second. Here in DOVE, the issue is worse with the DPS. In PortLand, it was just a database query to retrieve a PMAC address. In DOVE, the lookup searches for the address of the dSwitch, and corresponding policy rules and actions in order to determine the next action.

5.2 Core isolation for Cloud DC

In this section we introduce protocols using the Core isolation technique and with multi-tenant capabilities. We divide them into three categories depending on the Layer of the underneath network.

5.2.1 Core isolation for Layer 2 Cloud DC

The five protocols presented in this section required that the underneath network be a Layer 2 network in order to be used.

5.2.1.1 802.1ad (QinQ)

The IEEE 802.1ad standard [5] also known as "QinQ" is in fact an amendment to the IEEE standard 802.1Q. This amendment enables service providers to offer isolation to their customers' traffic. In addition to the VLAN information of the client, defined by the 802.1Q standard, this new 802.1ad standard defines another VLAN for the provider. The customer VLAN header is called the C-TAG (customer TAG) and is the inner header. The outer header is the S-TAG (Service TAG) for the provider. In Figure 19 we represent the two VLAN headers. The TPID0 field has a default value of 0x88A8 which is different than the default value (0x8100) of the 802.1Q standard. TPID1 is configured with the default value 0x8100. This differentiation indicates to the switch that there are two TAGs.

Thanks to the S-TAG header, the provider can manage only one VLAN for all the VLANs of one client. He is able to provide $2^{12} = 4096$ VLANS to each of his 4096 clients which results in $2^{12} * 2^{12} = 16777216$ different VLANs. If it is the solution chosen by the provider then the 802.1ad VLAN management is identical to the 802.1Q VLAN management because the provider only cares for the S-TAG header. It is the

client responsibility to manage his/her 4096 VLANs in his/her network.

However, most of the time, one client does not need 4096 VLANS and the provider has more than 4096 clients. So instead of using both TAGs separately, the provider adds both TAGs in order to have 16777216 VLANs. This way the management of such a solution is more complex than the 802.1Q solution, but yields greater scalability. For switching the frames, the switches have to recover the VID of both TAGS and verify in a database with up to 16777216 values instead of 4096. This implies more work to obtain the VLAN ID and consequently more time to verify the VLAN ID in the database, it also uses more memory space because of its increased size. It means more CPU, more memory, and more latency at each switch. Additionally, irrespective of the way both TAGs are managed, the overhead is increased by four bytes.

The advantage of the 802.1ad standard is that it raises the limit of VLANs possible from 4096, with 802.1Q, to 16777216, which should be sufficient for network growth during the next few years. If this new limit is still too small then there is the possibility to use the 802.1ad VLAN TAG stacking solution and add more VLAN TAGs to the header. However it is a non-standard solution and might results in overhead issues because each time we add a VLAN TAG, the header increases by four bytes for only 12 bits of VID.

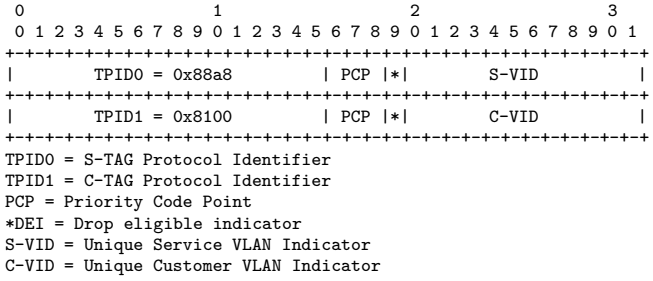


Figure 19: 802.1ad header

5.2.1.2 802.1ah (mac-in-mac)

The 802.1ah IEEE standard [6] was developed after the 802.1ad standard [5] in order to provide a method for interconnecting Provider Bridged Networks. This protocol is intended for network providers in order to attend to their needs for more service VLANs. This standard is also known as Provider Backbone Bridges (PBB) or "mac-in-mac". As the last name indicates, the idea is to add another MAC header on top of the existing MAC header. This new MAC header is added by a Provider Edge (PE) switch. This allows for the core network switches to only save the MAC of the PE switches, thus no MAC information of the client are used for switching inside the core network. All the mapping work is done by the PE switches and they are the ones responsible for encapsulating and decapsulating the messages.

The encapsulation is done by adding a new MAC header to the message. This new MAC header (Figure 21) is composed of two different MAC headers. The first one is the new header from the 802.1ah standard. This MAC header can be divided in two parts. The first part is the one with the Backbone components. The fields of this part are:

- MAC Backbone Destination Address (B-DA), 6 bytes long
- MAC Backbone Source Address (B-SA), 6 bytes long
- EtherType with a size of 2 bytes and a value of 0x88a8
- Priority Code Point (3 bits) and the Drop Eligible Indicator (1 bit)
- Backbone VLAN indicator (B-VID) with a size of 12 bits

After this Backbone part, the second part, called the Service encapsulation, is three bytes long and contains the following fields:

- EtherType with a value of 0x887e on two bytes
- Priority Code Point (3 bits) and the Drop Eligible Indicator (1 bit)
- Used Customer Address (1 bit) indicates if the customer address is valid or not
- Interface Service Instance Indicator (I-SID) with a size of 20 bits

With this new 802.1ah header we now have another MAC header for the provider to use. There are now 4096 VLANs possible with the B-VID. In each VLAN there are $2^{20} = 1048576$ supported services with the I-SID field. This could amount to a total of 4294967296 VLANs with only the new header. Additionally only the PE switches have to learn the customers' MAC addresses (C-DA and C-SA) and have to add and suppress the new 802.1ah header.

The 802.1ah standard is an evolution of the 802.1ad standard which is an evolution of the 802.1Q standard. Each new standard has added information in the header of the message. The Figure 20 shows the evolution of the 802.1 header. We can see that in order to increase the number of VLAN identifiers, the size of the header keeps increasing. This increase implies, as stated in the 802.1ad standard 5.2.1.1, that switches, at least the PE switches, use more CPU time to process the header and use more memory to save all the information of the VLANs.

5.2.1.3 Private VLANs

Private VLANs is a solution developed by Cisco and presented in "RFC 5517" [65]. This solution is based on the aggregated VLAN model proposed in "RFC 3069" [66]. The idea is to have a principal VLAN subdivided with secondary VLANs. The principal VLAN broadcast domain is therefore divided in smaller subdomains. A subdomain is defined by the designation

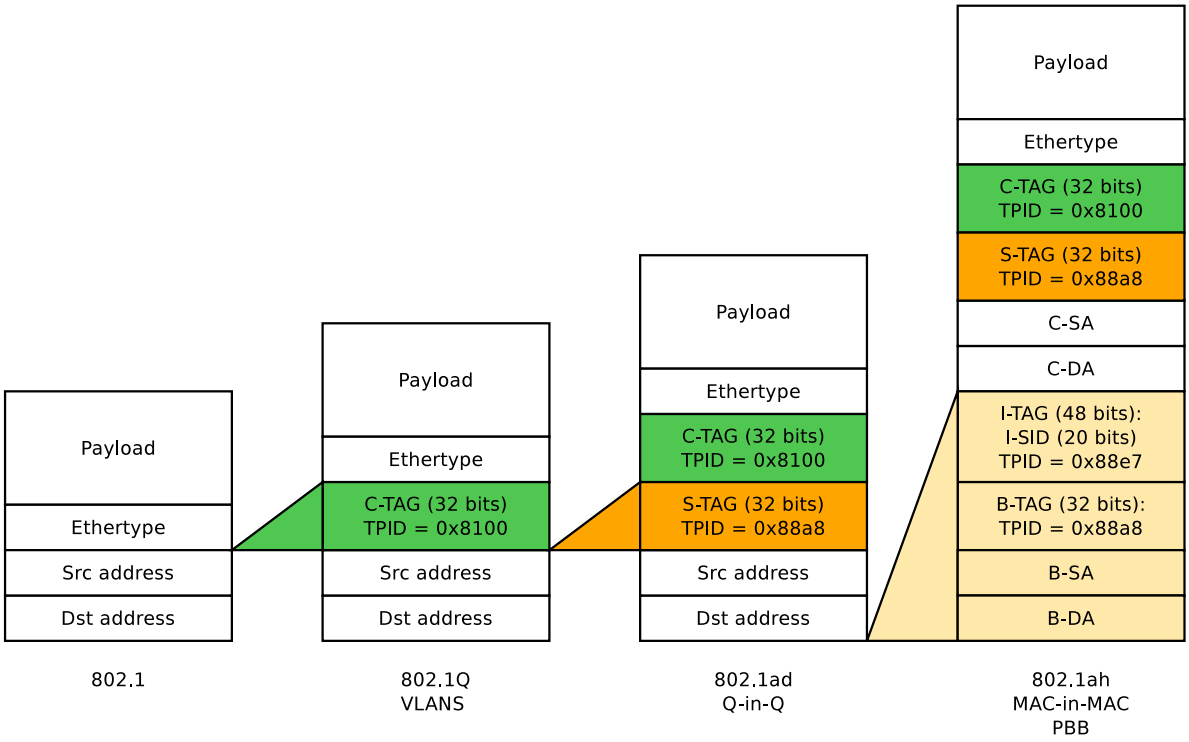


Figure 20: 802.1, 802.1Q, 802.1ad and 802.1ah frame formats (figure from "IEEE 802.1ah Basics"[64])

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             B-DA                              ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 B-DA (cont.)                 |    B-SA                           ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 B-SA (cont.)                                                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       TPID0 = 0x88a8          | PCP |*|         B-VID         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       TPID1 = 0x88e7          | PCP |*|!| RES |      I-SID      
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 I-SI (cont.)                  |     C-DA                      ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 C-DA (cont.)                                                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             C-SA                              ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 C-SA (cont.)                  |        TPID2 = 0x88a8         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| PCP |*|        S-VID          |        TPID3 = 0x8100         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| PCP |*|        C-VID          |  Ethertype of Payload         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Payload ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
B-DA = Backbone Destination Address
B-SA = Backbone Source Address
TPID0 = B-TAG Protocol Identifier
PCP = Priority Code Point
*DEI = Drop Eligible Indicator
B-VID = unique Backbone VLAN Indicator
TPID1 = I-TAG Protocol Identifier
!UCA = Used Customer Address
(0=Valid, 1=Not Valid)
RES = Reserved
I-SID = Interface Service Instance Indicator
C-DA = Customer Destination Address
C-SA = Customer Source Address
TPID2 = S-TAG Protocol Identifier
S-VID = unique Service VLAN Indicator
TPID3 = C-TAG Protocol Identifier
C-VID = unique Customer VLAN Indicator
```

Figure 21: 802.1ah header

of the switch's ports group. In [65] there are three port designations. These port designations are as follows:

1. Isolated port: An isolated port can not talk with an isolated port or a community port.
2. Community port: A community port belongs to a group of ports. Those ports can communicate between themselves and with any promiscuous port.
3. Promiscuous port: A promiscuous port can talk with all the other ports.

In order to create the subdomains within a VLAN domain, the VLAN ID is not enough. An additional VLAN ID is used. To refer to a specific Private VLAN, at least one pair of VLAN IDs is necessary. A pair of VLAN IDs is composed of one primary VLAN ID and of one secondary VLAN ID. The primary VLAN ID is the VLAN identifier of the whole Private VLAN domain. This scheme of VLAN pairing only requires the traffic from the primary and secondary VLANs to be tagged following the IEEE 802.1Q standard. It only uses a single tag at most, thanks to the 1:1 correspondence, between a secondary VLAN and its primary VLAN. The Private VLAN technique allows for a greater number of VLANs thanks to the recycling of VLAN IDs in secondary VLANs. It also allows for better addresses assignment in a domain because these addresses are shared between all the members of the private VLAN domain.

5.2.1.4 PortLand

The Portland architecture [9] is one that uses a centralized control plane, with the core network working

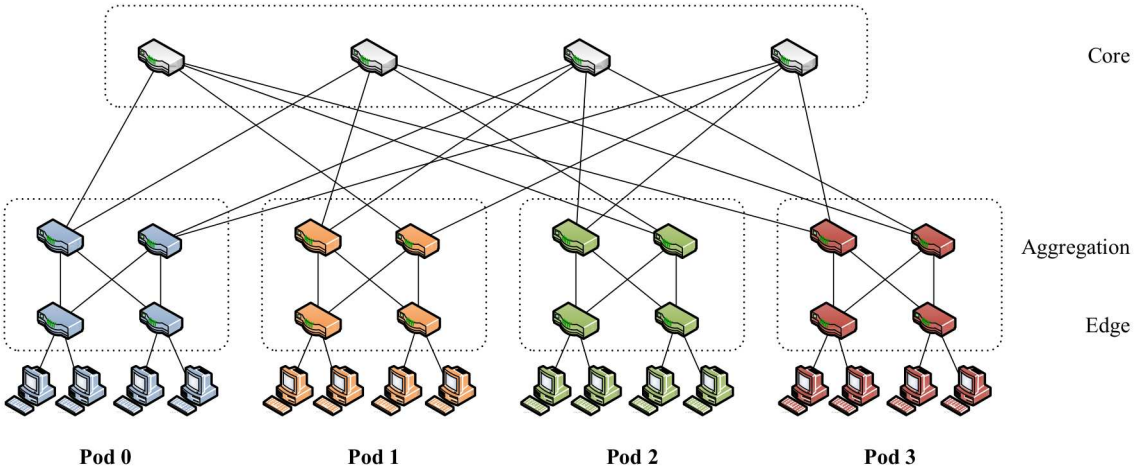


Figure 22: A fat tree topology (figure from PortLand paper [9])

at Layer 2. The topology of the network must be a multi-rooted fat-tree [67] as in Figure 22. To manage forwarding and addressing, a fabric manager is used. A fabric manager is a user process running on a dedicated machine which manages soft states in order to limit or even eliminate the need for administrator configuration. In addition to the fabric manager, Portland introduces new MAC addresses for each end host. This new MAC address, called Pseudo MAC (PMAC) address, encodes the position of the end host in the topology. The PMAC is 48 bits long and is composed of four parts.

1. pod (16 bits) is the pod number of the edge switch
2. position (8 bits) the position of the end host in the pod
3. port (8 bits) is the switch port number the host is connected to
4. vmid (16 bits) is used to differentiate the virtual machine on the physical machine

The PMAC is not known by the end host which keeps using its actual MAC (AMAC) for its packets. When an edge switch sees a new AMAC, coming from a connected machine, it has to create a new PMAC and map the IP address with the AMAC and the PMAC. It then has to announce the new mapping between the IP address and the PMAC address to the fabric manager. This way when an edge switch wants to forward a message with only the IP address, it will do an Address Resolution Protocol (ARP) request which will be processed by the fabric manager and receive the PMAC address in the answer. Edge switches are responsible for mapping the PMAC to the AMAC. They also have to replace the AMAC with the PMAC for outgoing packets and replace the PMAC with the AMAC for arriving packets. This way Portland can uses a hierarchical PMAC addressing with only the pod part, the first eight bits, used for forwarding the data through the core switches. Then at the aggregation switches, the position part is used and, at the next level, the edge switches uses the port part. Finally the last part, the vmid, is used by the server to know to which VM to deliver the packet.

Portland design implies that the fabric manager learns all the correspondences between PMAC addresses and IP addresses and uses this table to answer ARP requests from the edge switches. The edge switches then use the PMAC addresses they received to change the destination addresses. Since PMAC addresses are hierarchical, this enables switches to have smaller forwarding tables.

The fact that the fabric manager, a single machine, has to manage all the ARP traffic makes this architecture not able to scale. For example in a data center with 27,648 end hosts (not tenants) and each host makes 25 ARP requests per second, the fabric manager will need approximatively 15 CPU cores working non-stop to only manage the ARP requests.

5.2.1.5 SEC2 : Secure Elastic Cloud Computing

Secure Elastic Cloud Computing(SEC2) [10] is an architecture which uses a centralized control plane over a Layer 2 core network. In this architecture, the network is divided in one core domain and several edge domains (Figure 23). An edge domain possesses an identifier, the edge id (eid), which is unique among the edge domains. Each edge domain is connected to the core domain via Forwarding Elements (FEs) which manage address resolution and enforce policy rules. In an edge domain, tenants are isolated thanks to the use of VLANs. A tenant's subnet is identified by a unique Customer network id (cnet id). This implies that there is a 1:1 correspondence between a VLAN ID and a cnet id. In SEC2 there are only 4096 tenants in an

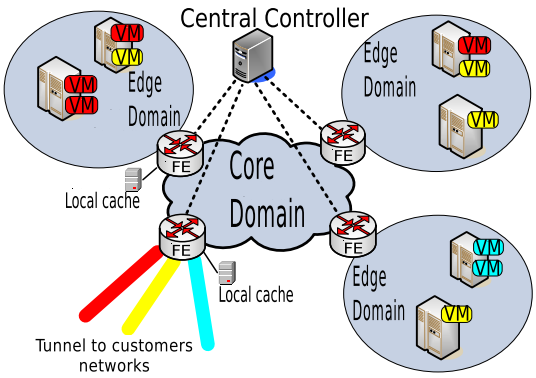


Figure 23: SEC2 Architecture (adapted from figure of SEC2 paper [10])

edge domain. However, the VLAN ID can be reused in a different edge domain so the maximum number of VLANs allowed does not limit the number of tenants. As SEC2 does not limit the number of edge domains, to increase the number of tenants, the solution is to create a new edge domain. A tenant's VMs are identified by the combination of the cnet id and their IP addresses. These IP addresses are freely chosen by the tenant and there is no restriction on them. The VM MAC address is then mapped to this combination (cnet id, IP). The MAC address is not needed by the FE to forward the packets because the pair (cnet id, IP) is unique.

For resolving addresses and enforcing rules, FEs must obtain the information from the Central Controller (CC) on an on-demand basis. When a VM wants to send a packet to a different VM, the VM only knows the IP address and therefore will do an ARP request. This ARP request is intercepted by the FE which looks in its local cache to see if the answer is present. If not, it sends a request to the CC which answers with information such as the MAC address of the receiver, the eid of the domain in which the receiver is located, and the VLAN ID of the receiver. The FE then answers the ARP request with the MAC address. The other information is saved in the FE's local database. When the packet reaches the ingress FE, the FE will encapsulate the packet with a MAC header. The destination address will be the eid previously received, and the VLAN number will be replaced by the one received from the CC.

For the CC to be able to answer the FE requests, the core, edge, and network information must be stored. The following mappings are maintained by the CC:

- VM MAC ↔ (cnet id, IP). To resolve the IP address of a VM and obtain its MAC address.
- VM MAC ↔ edge domain id (eid). To know in which edge domain the VM is located.
- eid ↔ FE MAC address list. To determine between which FE to establish the data tunnel if there are multiple FEs for a single edge domain.
- (cnet id, eid) ↔ VLAN id. To identify which VLAN to use in the receiver edge domain.
- cnet id ↔ rules and actions. To know if both tenants agree to communicate.

To allow inter sub-network communication, each tenant must have at least one public IP address stored in the CC.

Even if the design uses a centralized controllers, the CC can be distributed and the information can be divided per tenant. The author provides an example:

> For example, different customers can be assigned to different CCs by using Distributed Hash Table (DHT). Since the management and policy control of different customer networks are relatively independent, such partition does not affect the functionality of CC.

SEC2 provides tenant isolation and also address-space isolation thanks to the use of VLAN in edge domains. FEs enforce the rules and policies that are stored in the CC, which also prevents inter-tenant communication if it has not been previously agreed.

5.2.1.6 VNT: Virtual Network over TRILL

In [14] the authors propose a new technique of overlay network done over a Layer 2 network using the Transparent Interconnection of Lots of Links (TRILL) protocol [68, 69]. This overlay network is called Virtual Network over TRILL (VNT). In order to provide tenant isolation, VNT adds a Virtual Network Identifier (VNI) field in the TRILL header (Figure 24). The VNT header is composed as follows. The first 64 bits correspond to the basic TRILL header. They are followed by a block of 32 bits describing the criticality of the options. Then there are a reserved field of 18 bits and a flow ID field of 14 bits. The VNT extension is added as an option in the header, with the Type, Length, Value (TLV) format, and need a 64-bit block. The VNI field is 24 bits long and can differentiate approximatively 16 million tenants. A VNI is unique in the core network and is associated with one tenant. To apply this VNT extension to the packet, a new network component is introduced and is called a Virtual Switch (VS). A VS has to manage all the interfaces with the same VNI Tag (all the interfaces of one tenant). The provider administrator must link every new VMs of a tenant to the unique VS managing the VNI Tag associated with the tenant.

Tenants are free to use any Layer 2 or Layer 3 address they want in their virtual network. These addresses are not visible in the core network and cannot affect packet routing. The routing of the tenant data is done via two different routing techniques at different layers. The first routing, the virtual routing, is done at Layer 2, in the core network, through the VNI tag and some rules in the RBridges. A RBridge can only send a packet toward another RBridge or an end host if they share the same VNI tag. The second routing is done at Layer 3, and is dependent on both the tenant's network and the tenant's Layer 3 endpoint configuration.

This overlay technique allows an isolation of tenants' data thanks to a VNI Tag in the TRILL header. This VNI Tag is a 24-bit value which allows for approximately 16 million different tenants, which is better than the limit of 4096 VLANs. The VNI Tag is also used to establish unique tree topology for each virtual network associated to this VNI Tag using the intermediate system to intermediate system (IS-IS) protocol [70, 71]. This way the data tagged with a VNI will only be propagated along this tree and will not be send to other tenants' host, which ensure tenant isolation. Moreover the packets are routed based on the VNI tag in the physical network which isolates the space address of each tenant, so that a tenant can use Layer 3 and Layer 2 addresses.

However, VNT being based on TRILL, it is impossible to interconnect multiple data center without merging their control plane into one, resulting in losing each data center independence and increasing the broadcast domain. To prevent the merging of TRILL network when being interconnected and to keep each data center control plane independent, the Multi-Level TRILL Protocol with VNT (MLTP/VNT) solution has been developed. This solution, describe in [72], mostly improve TRILL scalability, thus allowing for a better use of VNT.

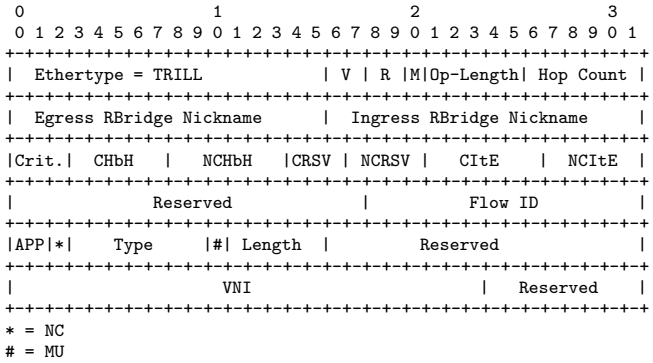


Figure 24: VNT Header

5.2.2 Core isolation for both Layer 2 and Layer 3 Cloud DC

The protocol introduced in this section uses the Core isolation technique and works over a Layer 2 and/or Layer 3 network.

5.2.2.1 VSITE

In [12], the VSITE architecture is proposed in order to allow enterprises (companies) to have seamless Layer 2 extensions in the cloud. In the paper, the tenants are considered to be exclusively enterprises that need to expand their networks. VSITE defines this extension as the collection of resources of an enterprise within a data center, and called it a virtual stub network, or vstub. The enterprise customer edge (CE) switch communicates with the cloud edge switch to exchange MAC information via an OTV-like protocol. For communication over the public network, VSITE uses Ethernet over IP (Ethernet over GRE (Section 4.1.1.1) or EtherIP [73] protocol) to transport data between the cloud edge switch and the CE. This OTV-like protocol uses a control plane protocol to exchange MAC addresses among sites which eliminates the cross-site MAC flooding. To suppress this flooding without modifying the VMs behavior, the hypervisor is tasked to intercept all the VMs' DHCP and ARP messages.

In VSITE there is no global VLAN ID assigned to enterprises but rather, local VLAN IDs at the data center edge location (between the switch and the VMs connected to it). The VLAN IDs are not statically assigned to enterprises. Therefore, the cloud edge switch has to map the VLAN ID of the enterprise to a locally-significant unique VLAN ID for the traffic from the enterprise's network to the VMs. The reverse operation has to be done at the hypervisor for the traffic from the VMs to the enterprise's network.

To ensure isolation of tenants' data, VSITE encodes the tenants' ID in the MAC addresses. The hypervisor will then ensure the traffic isolation by verifying that the VM receiving the packet belongs to the enterprise through the tenant ID in the MAC address. The hypervisor, by checking the tenant id, must either accept or drop the packet. The MAC address (48 bits long) is divided in two:

1. X bits for a tenant's id
2. 48 - x bits for the VM's id

Where X value is the administrator's choice.

The core network of the data center can be a Layer 2 or a Layer 3 network. In a Layer 2 network situation, the MAC-in-MAC encapsulation technique allows for a location MAC address (locMAC). With a Layer 3 network, the packet is encapsulated in an IP packet with a location IP address (locIP). The locIP or locMAC are location addresses assigned to a VM. Each VM possesses a location address which allows the separation of its name and location. The name of the VM is the IP address assigned by the enterprise, the enterprise IP (entIP). The location of the VM is indicated by the IP address, or the MAC address, of the switch to which the VM is logically connected. However this logical connection must be done via Ethernet. This location address is used to route the packet in the core network. All locIP (or locMAC) addresses are stored in a directory server. Because of this, a VM or data center edge has to send a lookup request to the directory server to retrieve the locIP of the destination if the information is not already in its local cache. The directory server maintains the mapping between entIP and pertinent information including locIP, MAC, and potentially, a VLAN ID.

The VSITE architecture has a centralized control plane and relies on hypervisor security to provide protection against MAC address spoofing, a VM impersonating a different VM, or a DDOS attack from a VM. It also uses Core isolation protocols in both its edge domains (VLANS) and its core network (MAC-in-MAC or IP-in-IP). However, data transported over

the public network is not protected.

5.2.3 Core isolation for Layer 3 Cloud DC

The protocols introduced in this section use the Core isolation technique and require a Layer 3 network.

5.2.3.1 VRF: Virtual Routing and Forwarding

Virtual Routing and Forwarding (VRF), described in [74], is a technology included in routers. This technology allows a router to have multiple instances of a routing table to exist and work at the same time. With these multiple routing tables it is possible to segment the network and separate users in different routing table instances. The Customer Edge (CE) router does the local routing and then exchanges the information with the Provider Edge (PE) router. The PE router creates a VRF instance with the information from this CE. This new VRF instance is used by the PE for every packet to and from this CE. For each CE corresponds a VRF instance in the PE. This allows for the creation of a Virtual Private Network (VPN). The PE router creates at least one routing table instance for each VPN. Then the PE routes the packet from each VPN by searching in the corresponding VRF instance. The separation of the VRF instances provides a secure access to all the devices in a specific VRF instance. It also allows the use of the same or overlapping IP addresses without conflict. The VRF method is intended to help the ISPs provide private networks to their clients while using the same physical infrastructure. These private networks, over shared infrastructures, are called Virtual Private Networks (VPNs). It is as if the whole network of each client is tunneled. This solution is based on MPLS for the routing of the packets in the core network. It is not used in data centers.

5.2.3.2 VXLAN: Virtual eXtensible Local Area Network

Virtual eXtensible Local Area Network (VXLAN), detailed in "draft-mahalingam-dutt-dcops-vxlan-09" [7], is being developed in order to expand the VLAN method and remove the VLAN limit of 4096. VXLAN allows overlaying a Layer 2 network on a Layer 3 physical network. Such tunnels begin and end at VXLAN Tunnel EndPoints (VTEPs). The ingress VTEP encapsulates the packet and adds a VXLAN header of 8 bytes to the packet. The header (Figure 25) is composed of 8 bits of Flags, 1 bit for each flag, with the I flag, the 5th one, set to 1 for a valid VXLAN Network ID (VNI). Then comes a 24-bit long reserved field. It is followed by the VXLAN Network Identifier (VNI) field, 24 bits long, used to identify which individual VXLAN overlay network the packet belongs to. Finally the header ends with another reserved field with a length of 8 bits. Once the VXLAN header is added, the packet is then encapsulated within a UDP header. This UDP header must use the value 4789 as destination port. This value has been assigned by the IANA as the VXLAN UDP port. The source port is provided by the VTEP. Both headers are removed at the egress VTEP.

One advantage of VXLAN is that it expands the VLAN technology with a larger number of VXLAN possible. On the other hand, VTEPs must not fragment encapsulated VXLAN packets and if such a packet has been fragmented along the path it must be silently discard by the egress VTEP. As UDP is used to encapsulate the data and that a packet must be discarded silently, the source does not know that its packet has been discarded. There is no security measure so it is recommended to use IPSec to add security mechanisms.

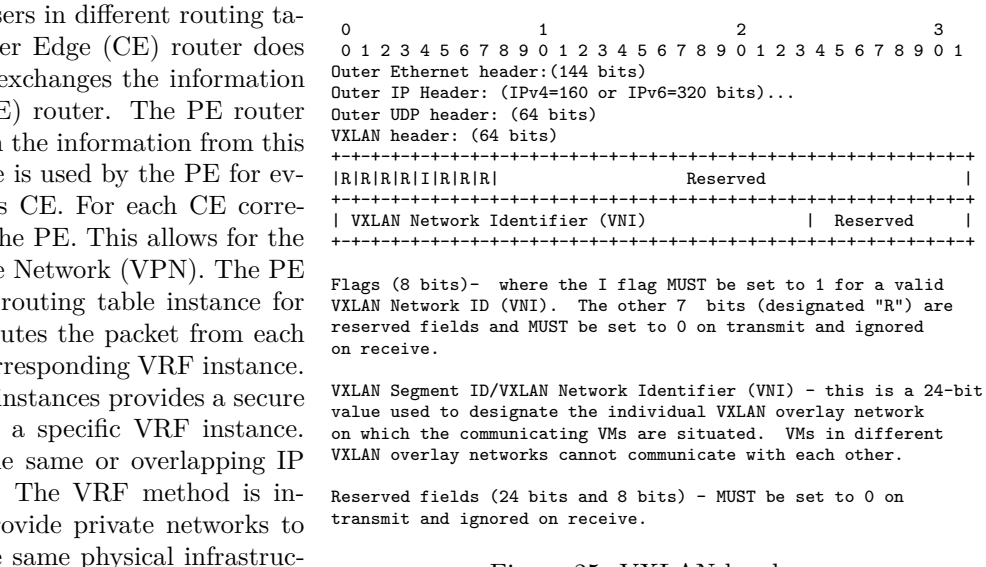


Figure 25: VXLAN header

6 Network isolations provided by several cloud tools

This section regroups a succinct list of several tools used in cloud deployment which provide network isolation while relying on already established tunneling protocols.

6.1 Cisco guide

In [19] tenant isolation is done thanks to path isolation and device virtualization. Path isolation is as if the packet from this path went through a tunnel over the network (Figure 3). The device virtualization is achieved by creating virtual devices such as VMs or virtual switches.

Path isolation is done thanks to several techniques which provide an independent logical path over a shared infrastructure. To create these paths, two technologies, over different layer, are mainly used:

1. Layer 2 separation with VLAN
2. Layer 3 separation with VRF

Separation on the Layer 2 is done thanks to Virtual Local Area Network (VLAN) presented in Section 4.2.1.1.

At the Layer 3, the separation is done with Virtual Routing and Forwarding (VRF) presented in Section 5.2.3.1.

This solution works for small multi-tenancy clouds, with less than 4096 tenants, which corresponds to the VLAN limit. However in today's cloud data center with virtualization, there is a need to host more than 4096 tenants so this solution is not scalable enough.

In [20] tenant isolation is done with additional logical mechanisms such as virtual Network Interface Controllers (vNICs), IPsec or SSL Virtual Private Networks, packet filtering, and firewall policies.

The Security Architecture for the Internet Protocol (IPsec) is a security architecture focused on IP-Layer security, as mentioned in "RFC 4301" [75]. The Secure Sockets Layer (SSL) Protocol is designed to provide privacy and reliability between two communicating applications as described in "RFC 6101" [76].

6.2 Openstack software

Openstack manages tenant isolation thanks to the use of VLANs or Layer 2 tunneling with GRE, as stated in the OpenStack Security Guide [77]. For the Layer 2 tunneling, OpenStack Networking currently supports GRE (Section 4.1.1.1) and VXLAN (Section 5.2.3.2). However the support of VXLAN is added via the Neutron Modular Layer 2 (ml2) plugin.

The security guide explains that "the choice of technology to provide L2 isolation is dependent upon the scope and size of tenant networks that will be created in your deployment". Indeed if the environment has a large number of Layer 2 networks (ie. more than 4096 tenants with each their own sub-network), the VLAN limit could be reached and no more tenants could be added. Therefore it is better to use the tunneling method with GRE or VXLAN.

6.3 OpenFlow controller

OpenFlow [78] has a centralized control plane and uses an OpenFlow controller. This OpenFlow controller has a global view of the network and decides what is best for the network. It then sends its decisions to the compatible OpenFlow network switches. These switches can be hardware or software. Belonging to this last category, Open vSwitch is a virtual switch for hypervisors. It provides network connectivity to virtual machines. Open vSwitch works via a flow table which defines rules and actions for each flow.

In order to isolate the flows from each tenant, OpenFlow uses VLANs (Section 4.2.1.1) and GRE (Section 4.1.1.1) as a tunneling protocol. However Open vSwitch can also manage the Stateless Transport Tunneling (STT) protocol (Section 5.1.3.3) inherited from Nicira development.

6.4 Amazon's Virtual Private Cloud (VPC)

Amazon's VPC [79] is a proprietary solution so we only have scarce information. It provides a Layer 3 abstraction with a full IP address space virtualization. It is possible to create up to 200 sub-networks per VPC and to have up to 5 VPCs per customer. In order to communicate with Amazon's VPN you must have:

- The ability to establish IKE Security Association using Pre-Shared Keys (RFC 2409) [80].
- The ability to establish IPSec Security Associations in Tunnel mode (RFC 4301) [75].
- The ability to establish Border Gateway Protocol (BGP) peering (RFC 4271)[81].
- The ability to utilize IPSec Dead Peer Detection (RFC 3706) [82].
- The ability to adjust the Maximum Segment Size of TCP packets entering the VPN tunnel (RFC 4459) [83].
- The ability to reset the "Don't Fragment" flag on packets (RFC 791) [84].
- The ability to fragment IP packets prior to encryption (RFC 4459) [83].

However we do not know what happens inside the VPC. We could not find any documentation of how it is implemented either, and hence cannot comment on its isolation techniques.

7 Comparison

In this section, a comparison between fifteen of the solutions (protocols and architectures) previously introduced in Section 5 will be presented. The solutions being compared are: 1. LISP, 2. NVGRE, 3. STT, 4. 802.1ad, 5. 802.1ah, 6. VXLAN 7. Diverter, 8. PortLand, 9. SEC2, 10. BlueShield, 11. VSITE, 12. NetLord, 13. VNT, 14. VL2, 15. DOVE. This comparison is made by using six criteria. The first criterion is the complexity of use of the solution. The second is the overhead induced by each solution. Then we compare the solutions' capability to migrate VMs, followed by a comparison of their resilience. The fifth criterion is scalability, and finally, we study if it is possible and easily manageable to have multiple data centers.

7.1 Complexity comparison

To determine the complexity of a technique we take into account six criteria:

1. Centralized/Distributed control plane
2. Network restrictions
3. Tunnel configuration and establishment

4. Tunnel management and maintenance
5. Multi-protocol
6. Security mechanism

Table 2 summarize this comparison.

7.1.1 Centralized/Distributed control plane

The first criterion is if the technique has a centralized control-plane or not. Among the presented solutions, eight of them have a centralized control-plane. These solutions are LISP, PortLand, SEC2, BlueShield, VSITE, NetLord, VL2 and DOVE. They all possess a key component which is used to resolve addresses. PortLand, SEC2, BlueShield, VSITE, VL2, and DOVE architectures use this centralized controller to also maintain the rules allowing tenant traffic isolation. However these architectures mitigate the consequences of a failure of this key component through replication. This replication increases complexity because those replicas must all possess the same information, which implies a synchronization of these components.

The other architectures do not possess such a key component and possess a distributed control-plane. The failure of a single device will not compromise the entire architecture. However they need to store redundant, and sometimes unused, information in every switch or VM to do the address resolution. Whereas with a centralized control plane approach the switches only get the information they need from the key component on an on-demand basis.

The centralized control plane design also has other issues. For example, with the PortLand architecture, in a data center with 27.648 end hosts (not tenants), and each host makes 25 ARP requests per second, the fabric manager will need approximatively 15 CPU cores working full-time just to manage the ARP requests. In addition, in PortLand, the fabric manager only manages ARP requests, whereas the other architectures, which also use a centralized controller, additionally manage rules and policies, which increases the workload of this component. To mitigate this increased workload, redundancy of the centralized controller and local caches in switches are used. There is a need for synchronization between all these components. To prevent data routed with outdated information from the local caches to attain a tenant network, SEC2, Blueshield, VSITE and VL2 use local modules (FEs, Echelon VMs, Hypervisors, and VL2 agents respectively), to enforce rules, and to drop packets that do not conform to these rules.

7.1.2 Network restrictions

Only three solutions, 802.1ad, 802.1ah, and VSITE, do not impose restrictions on the underlying network. Then there are architectures that impose a Layer level on the network such as SEC2 and BlueShield which need a Layer 2 network, or DOVE and LISP which need a Layer 3 network. Three architectures need a specific topology such as a Layer 2 multi-rooted fat tree topology for PortLand, a Layer 3 Clos topology for VL2 and a flat Layer 2 network for Diverter. Two protocols (NVGRE and VXLAN) require that the Layer 3 underlying network does not fragment packets and for VXLAN that there is an IGMP querier function enabled. For NetLord, the network must be a Layer 2 network but with edge switches supporting IP routing, which is why NetLord is put as a Layer 2-3 architecture. VNT is using the TRILL header so it needs the edge switch to support the TRILL protocol and a TRILL or Layer 2 core network. Finally STT is the one that is the most tricky, as it only needs a Layer 3 network but it uses a modified TCP header. STT needs to be allowed to transit in all the middle boxes of the network.

In this category we started with solutions that do not impose restrictions on the underlying network, continued with architectures needing a specific Layer level, and finished with architectures needing a very specific topology or protocol. In order to use these latter architectures, the underlying network might have to be heavily modified, which could be difficult or even impossible to do on already established infrastructures.

7.1.3 Tunnel configuration, establishment, management and maintenance

BlueShield, Diverter, and PortLand are exceptions because they do not use encapsulation protocols. Instead, to provide isolation, BlueShield uses a Directory server and ARP requests to verify whether a VM can communicate with a different VM. If communication is permitted in the Directory server rules, then the address resolution is possible and the directory server responds to the ARP request. If not then the directory server does not reply and the address resolution fails, therefore data is not sent between the VMs. For Diverter the VNET replaces the VM's MAC address with the server's MAC address, and verifies whether the communication is allowed.

Eight solutions (LISP, NVGRE, STT, VXLAN, SEC2, NetLord, VNT and VL2) among the other twelve use an implicit tunnel configuration and establishment for the core network part. They do not exchange messages or make reservation to establish the tunnel.

The others three use an explicit tunnel configuration. Those three solutions are the same three which does not impose restriction on the network. 802.1ah and 802.1ad both use the GARP VLAN Registration Protocol (GVRP), which is a GARP application, in order to distribute the VLAN registration information in the network. The last of the three, VSITE, uses the MPLS VPN protocol on the public network, which uses an explicit tunnel configuration.

Even if tunnel establishment could potentially be implicit, for twelve of the solutions, tunnel maintenance and management must still be explicit. LISP uses addresses mapping in its ITR (Ingress Tunnel Router) and ETR (Egress Tunnel Router). 802.1ad and 802.1ah both use join and leave messages sent by end stations

and bridges. VXLAN also uses join and leave messages. However they are sent by VTEPs with the goal of keeping the distribution tree of each VNI updated to reach all the clients of this VNI. Diverter, SEC2, BLueShield, VSITE, and VL2 each maintain rules for isolation which need to be managed and updated according to tenant requirement, or in case of VM migration. To enforce these rules, they all defined agent modules detailed in Section 7.2.3. PortLand uses soft states to maintain the tunnel and VNT uses temporary forwarding database in its RBridges. NetLord uses a SPAIN agent to manage the tunnel.

The two which do not have tunnel management are NVGRE and STT. For DOVE, tunnel management is dependent upon the encapsulation protocol.

7.1.4 Multi-protocol and security mechanism

Only PortLand and VNT are multi-protocol. NVGRE and BlueShield accept protocols from the second Layer because they use the MAC address to forward the packet at the last hop to the correct VM. STT, 802.1ad, 802.1ah, VXLAN, VSITE, and NetLord solutions also use the MAC address for delivering the packet but the protocol must be Ethernet. LISP, Diverter, SEC2 and VL2 use the IP address instead of the MAC to deliver packets, therefore the protocol must be IP.

About security mechanisms, these architectures do not define any encryption, authentication or integrity verification techniques. However five of them (Diverter, SEC2, BlueShield, VSITE, VL2) have security mechanisms for tenant isolation thanks to their agents and directory services which enforce pre-established isolation policies. In BlueShield, the Echelon VM (the agent) can also be associated with a firewall to improve security by processing the traffic only after it traverses the firewall.

7.2 Overhead comparison

To determine overhead we list the encapsulation headers, messages, and components used by each architecture. All these elements are summarized in Table 3.

7.2.1 Encapsulation

Among the fifteen solutions presented, one (BlueShield) does not use any encapsulation or address rewriting, two (Diverter and PortLand) do not use encapsulation either, instead rewriting the address of the packet. For Diverter, the MAC address of the virtual machine is replaced by the MAC address of the physical remote node. In PortLand, the Actual MAC (AMAC) is replaced by the Pseudo MAC (PMAC). The others use encapsulation.

LISP defines its own header of 64 bits and also uses both UDP and IP headers as outer headers. So when the packet enters a LISP tunnel, a header of 288 bits, for the IPv4 version, or a header of 448 bits, for the IPv6 version, is added to the packet.

NVGRE has its own header (64 bits) and then encapsulates the packet within an outer IPv4 header (160 bits) and an outer MAC Ethernet header (144 bits), making a total of 368 bits in IPv4 or 528 bits in IPv6.

STT encapsulates Ethernet messages with a STT header(144 bits) then with a TCP-Like header(192 bits), an outer IP header(IPV4: 160 bits, IPV6: 320 bits) and finally an outer Ethernet header (144 bits) for a total cost of 640(IPv4) or 800(IPv6) bits.

VXLAN also defines its own 64-bit header for encapsulation, and does not rewrite addresses.

802.1ad modifies the MAC header by adding an S-TAG of 32 bits, after both MAC addresses, followed by a C-TAG of 32 bits. After these modifications, the header size increased by 64 bits. The 802.1ah header is increased by a complete MAC header. As such we have a MAC destination (64 bits), a MAC source (64 bits), a B-TAG (32 bits), and an I-TAG (48 bits) for a total of 176 bits. However it is possible to use the 802.1 standard with 802.1Q frames or with 802.1ad frames. In the first case we have to add the 802.1Q header which is 32 bits long and so the new header is 208 bits long. In the second case, the new header is increased by the S-TAG and the C-TAG from the 802.1ad standard and is now 240 bits long.

DOVE does not specify an encapsulation protocol but proposes to use one of the three previously presented protocols (NVGRE, VXLAN, or STT). SEC2 and VSITE use MAC encapsulation with a header of $18 * 8 = 144$ bits. In addition VSITE changes the destination address with a locMAC address. However VSITE can be deployed over a Layer 3 network so instead of MAC encapsulation it can use IP encapsulation (160 bits in IPV4, 320 bits in IPV6) and replace the destination address with a locIP address. VL2 also uses an IP encapsulation and rewrites the destination address with a LA (Location Address). As VNT is deployed over a TRILL or a MLTP network, it uses a modified version of the TRILL header. This modified version is 192 bits long. NetLord uses both MAC and IP headers to encapsulate the data, which creates an overhead of 304 bits in IPv4 and 464 bits in IPv6. NetLord is the solution which has the largest header overhead.

7.2.2 Messages

Both NVGRE and STT do not exchange messages. This is explained by the fact that these two solutions are tunneling protocols and they only define an encapsulation technique. Then, BlueShield only uses lookup requests, from the BlueShield agent to the Directory Server, to resolve addresses and suppress ARP broadcasts. Diverter does not provide a specific type of message, and resolves addresses using the ARP protocol.

802.1ad and 802.1ah both use the Generic Attribute Registration Protocol. VXLAN requires its VTEP to manage the distribution tree of each VNI by sending join and leave messages for each VNI.

VNT is based over a TRILL or a MLTP network, which implies the use of TRILL messages. As the Control plane in TRILL and in MLTP is based on the IS-IS protocol, in order to route frames it uses SPF (Short Path First) tree topology generated by Link State PDU (LSP) messages.

DOVE, like all the other solutions with a centralized control plane (PortLand, SEC2, BlueShield, VSITE, NetLord, and VL2), must store the mapping between VMs' addresses and dSwitches' addresses in the DOVE Policy Service (DPS). All the servers of the DPS have to exchange information to be synchronized. DOVE dSwitches must also make unicast requests to retrieve the information from the DPS.

SEC2 needs communication between Forwarding Elements (FE) and the Central Controller (CC) to first save the mapping between the VM addresses and the FE addresses. Then the FEs have to intercept the ARP requests of the VMs and convert these to unicast lookup requests sent to the CC. However, it is possible that there are more than one CC which must possess the same information. This implies the need for synchronization messages between CCs. The other possibility is that every FE sends information to all CCs. Additionally, SEC2 uses VLAN, so it needs to use the Generic Attribute Registration Protocol.

NetLord is based on the Diverter model for resolving addresses, but it also needs unicast messages between SPAIN agents and the repository to obtain the table that maps destinations and sets of VLANs. In case of a topology change, new messages must be sent to update this table.

VL2 uses registration messages to store the association between AAs and LAs in the directory system. To resolve addresses the VL2 agents send lookup requests to the directory service. However there might be multiple directory services so they must be synchronized. In addition, for LA address assignment, VL2 uses an IP-based link state routing protocol.

VSITE uses an OTV-like protocol to exchange MAC addresses between CEc (Customer Edge cloud), CEt (Customer Edge tenant) and the directory server. This protocol allows the elimination of the cross-site MAC learning flooding. As an architecture with a centralized control plane, VSITE has to perform Directory lookups for address resolution. For these queries, unicast messages are sent from the CEc or VSITE agent to the Directory server. For address resolution between CEt and CEc they exchange MAC reachability information using OTV control plane protocol.

PortLand has four functionalities that need messages. The first is the registration of new source MAC address, as seen at the ingress switch, at the fabric manager to save the mapping between the PMAC, the MAC, and the IP addresses. To resolve addresses, PortLand intercepts the ARP requests of the VMs and converts them in unicast lookup requests to the fabric manager. However, if the fabric manager cannot answer one of the lookup requests, it must broadcast an ARP request to all end hosts. In addition, PortLand switches periodically send a Location Discovery Message (LDM) out of all their ports, both to identify their position, and to perform health checks. Finally, it is possible to have fabric manager synchronization messages in the case of redundant fabric managers.

LISP uses five different messages. The LISP Map-Request is used to request a mapping for a specific EID, to check the reachability of an RLOC, or to update a mapping before the expiration of the TTL. However in [2], it is RECOMMENDED that a Map-Request for the same EID-Prefix be sent no more than once per second. The second message is a LISP Map-Reply which is the answer to the LISP Map-Request. This message returns an EID-prefix whose length is at most equal to the EID-Prefix requested. The LISP Encapsulated Control Message(ECM) contains the control packets of the xTRs and also the mapping database system from [85]. Also defined in [85], the messages LISP Map-Register and LISP Map-Notify are used to manage the xTR/EID-Prefixes associations.

7.2.3 Components

Every solution presented in this survey carries at least one new networking dependency. Only five of them define exactly one component. LISP needs an xTR component. This component is the device at each end of the tunnel. It must function both as an Egress Tunnel Router (ETR) and as an Ingress Tunnel Router (ITR). Additionally the xTR needs to work as a Proxy ETR (PETR) and as a Proxy ITR (PITR) in order to connect LISP to non-LISP sites. Diverter introduces VNET, a software module which resides within the host OS on each physical node. Then NVGRE, VXLAN, and STT need NVGRE Endpoints, VXLAN Tunnel Endpoints (VTEP), and STT Endpoints. All three are modules in switches, servers, or hypervisors, which encapsulate and decapsulate the packets.

With two new components each, PortLand, VNT, VL2, and DOVE belong to the same group. Portland, VL2, and DOVE use a centralized controller. PortLand defines a Fabric Manager, VL2 a Directory System, and DOVE a DOVE Policy Service. To apply the rules stored in these centralized controllers, PortLand uses edge switches which must be able to perform MAC to PMAC header rewriting. VL2 defines an VL2 agent added to the hypervisor and DOVE defines dSwitches which are the edge switches of the DOVE overlay network. On the other hand, VNT does not use such a centralized controller. Instead VNT uses RBridges, which provide the advantages of Layer 2 (Bridges), Layer 3 (Routers), and Virtual Switches(VS). A VS is dedicated to host all interfaces tagged with a particular VNI corresponding to a tenant ID.

SEC2 and VSITE define three components each. SEC 2 uses a Central Controller (CC) but this device could possibly be on several servers for redundancy or load balancing. To enforce the rules of the

CC, Forwarding Elements (FEs) are introduced. A FE is a switch that intercepts ARP requests from VMs and encapsulates data if necessary. The third component is a web portal, where each customer can set up security policies for their network, which then translates them into policy settings saved in the CC. VSITE uses a Directory server to save the addresses associations. It presents a component called CEc (Customer Edge cloud) which is in the cloud data center. This CEc encapsulates the Ethernet frames received from the tenants' private networks with an IP header. The IP destination address is the address of the Top of the Rack switch which hosts the Ethernet frame's destination device. This device is in the tenant vstub. This CEc prevents the overlapping of VLAN IDs from multiple companies by translating this VLAN ID into a locally unique one. For an Ethernet frame from cloud VMs, the translation is done at the VSITE agent in the hypervisor. The Ethernet frame is then encapsulated with an outer IP header.

NetLord uses NetLord Agent (NLA) implemented at each physical server to encapsulate the data with an IP header and then with an Ethernet header. To do load balancing when sending packets, the NLA uses a SPAIN agent that is implemented in the NLA. The third component, an edge switch, is not really a new one. However, this edge switch must be able to read the IP header of the packet. And the last new component is a configuration repository which maintains all the configurations of the tenant virtual networks.

As the other architectures with a centralized control plane, BlueShield uses a Directory Server and an agent, called BlueShield Agent, to enforce the rules of the Directory server. For security measures, an Echelon VM is introduced and is in fact a VM that scans the traffic to apply added actions such as sending the traffic flow through a firewall. To suppress ARP flooding, a virtual switch is installed in the server and converts the ARP requests to unicast directory lookups. An additional component, 'ebtables' firewall, has been used to block all broadcast and multicast traffic.

Both 802.1ad and 802.1ah require that all the devices of the network adhere to their respective standard. These two solutions being IEEE standards, the devices are not modified by the network administrator but by the manufacturers of these devices in order to comply with the standard.

7.3 Migration of VM comparison

The migration of a VM is an important task in a virtualized data center. When a server needs to be shut down for maintenance, the VMs on this server must not be stopped so they have to be moved to another server. If a client's location changes, it might be interesting to move their VM accordingly. VM migration can be done in two different ways, an offline migration or a live migration. The offline migration will stop the service by terminating the session and establishing a new session once it has finished migrating. The one we are interested in here is the live migration which allows for a continuity of service and session even while the VM is being moved. We summarize this comparison in Tables 4 and 5.

LISP RFC [2] defines five types of mobility, however only three of them concern endpoint migration:

1. Slow endpoint Mobility: An endpoint migration without session continuity uses "RFC 4192" [86].
2. Fast Endpoint Mobility: An endpoint migration with session continuity.
3. LISP Mobile Node Mobility: An xTR migration.

Among these three types of mobility, only the last two are of interest to us for this comparison. For the Fast Endpoint Mobility, the solution is to use the technique of home and foreign agents. The home agent, the endpoint original agent, redirects traffic to the foreign agent of the network to which the endpoint moved. This technique is defined in "RFC 5944" [87] for IPv4 and in "RFC 6275" [88] and "RFC 4866" [89] for IPv6. However the last migration, the LISP mobile node mobility, allows the migration of device without the need of agents. As the device is itself an xTR, it can use topologically independent EID IP addresses. Thus it only has to register itself at the MAP-servers and Map-Resolvers of the network. This last solution is explained in [90].

NVGRE is an encapsulation and tunneling protocol. Its original goals were to increase the number of VLAN subnets; the VLAN technology being limited to 4096 subnets, and to achieve a multi-tenant environment. [91] states that NVGRE achieved its goals. However, NVGRE left the management of VM migration to IP because of its use of a UDP header. In order to improve the management of VM migration, the draft [92] defines new extensions for the control plane of NVGRE. Among these extensions, one is interesting for host migration. The REDIRECT message is in fact the original message, or at least the maximum data of the original message a REDIRECT message could contain, sent back to the sender. This REDIRECT message is sent by the old NVE, where the endpoint was hosted before migrating. The data of the returning packet starts with the address of the new NVE managing the endpoint. This address is 32 bits long and is the first information in the payload of the packet. Then follows a copy of as much data as possible of the original message. This way the sender now knows the address of the new NVE. However it is not specified how long the old NVE must maintain the information of the VM migration.

As NVGRE, STT is an encapsulation and tunneling protocol. However as opposed to NVGRE, there are no STT mechanisms for VM migration. STT is working with IP and it uses IP mechanisms to manage VM migration, but it must also use the IP mobility mechanism of STT.

802.1ad and 802.1ah manage VM migration the same way thanks to the GARP VLAN Registration Protocol (GVRP). When a VM moves, it must send a GVRP

message to the closest switch in order to indicate that the VLAN announced in the message is of interest for this machine. This way the VLAN tree will reach the VM. As the VM does not change its IP address, the connection is not lost. However, in order to keep the connection, the VLAN must be deployed in the destination device ahead of time in order to already have the distribution tree of this VLAN reach this device. Even with this advance deployment the migrating VM must stay in the same Layer 2 network.

VXLAN is a tunneling technique that allows a VM to migrate even to another network across a Layer 3 network. To do so, VXLAN uses join and leave messages destined to VTEP in order to indicate which distribution tree the VTEP must associate with. As for 802.1ad or 802.1ah, in order to have session continuity, the destination VTEP must be informed ahead of time that it must join the distribution tree requested by the migrating VM.

Diverter was designed to increase isolation between tenants' networks without degrading the overall performance of the network. A VM uses a virtual IP address which is created based on the Farm and Sub-network it belongs to. This IP address is formatted as follows:*10.Farm.Subnet.Host*. So in Diverter, all the VMs of one client belongs to the same sub-network and this sub-network is in one Farm. If a VM migrates to another server it means that this new server will now have to extend the Farm and the Sub-network. However, to discover the mapping between IP and MAC addresses, the VNET ARP engine uses multicast ARP, so the migration of the VM is not detected at the beginning of the migration, but only when the VNET ARP engine sends an ARP query, and the response has been received from the new server. If an existing connection was established between the VM pending migration and another VM, this connection will be interrupted at the beginning of the migration. The VNET of the non-migrating VM will continue to associate the MAC address of the old server, where the migrating VM was hosted, to the traffic of this session. This traffic will be lost until the VNET ARP cache entry times out and the VNET does an ARP query to retrieve the new MAC address. Nevertheless, since the IP address stays the same there could be no interruption of session even if, during the migration and until the VTEP learns the new MAC address, the traffic is lost. The session continuity depends on two parameters: The ARP cache entry timeout, and the TCP timeout. If the first one is longer than the second, then the session is lost and there is no live migration. On the other hand, if the TCP timeout is longer than the ARP cache entry timeout, the new MAC address will be retrieved before the end of the TCP session thus having session continuity and live migration.

PortLand defines Layer 2 messages to be sent when a VM migrates. Additionally, VMs' IP addresses remain unchanged during the migration. Thus, PortLand manages live migration as well as session continuity. These messages are sent only after the VM migration. The first message is a gratuitous ARP, sent by the migrated VM, which contains the new IP to MAC address mapping. It is forwarded to the fabric manager which then forwards an invalidation message intended to the old switch of the migrated VM. Upon reception of this message, the old switch sets up a flow table entry to trap the packets destined for the VM which has migrated. Additionally, when such packet is received at the old switch, it sends back a unicast gratuitous ARP to give the new PMAC address of the migrated VM. Optionally, to limit the loss of packets, the old switch can transmit the trapped packet to the VM.

As SEC2 architecture is composed of multiple edge domains, where the VM are hosted, and one core domain, which interconnect these edge domains, there are two ways a VM can migrate. First the VM stays in the edge domain. The migration consists of transferring a dynamic VM state from a source to destination hosts. Once the transfer is complete, a gratuitous ARP message is sent by the destination host to announce the VM's new location. This ARP message is sent only within the VLAN inside the edge domain, and can only reach hosts in this VLAN. However, if the VM migrates to a different edge domain, then the Central Controller (CC) has to update the VM's location in its table, including both eid and VLAN id. Since the IP address is not modified and the MAC address change is induced by the gratuitous ARP, then the migration is done without losing the session continuity and so SEC2 can perform live migration.

In [11], it is said that BlueShield allows live migration of protected VM. It is possible because BlueShield uses a Layer 2 core network and addressing scheme. As the IP address of the VM is untouched, the continuity of the session is preserved. However the process of migrating a VM is not clearly defined in the solution. We can guess that as each VM need to have a BlueShield agent which manages ARP queries, this same agent must warn the directory server of the migration of the VM and provide the new MAC address. This way the directory server informs the other VMs of the new address of the migrated VM. The echelon VM could manage the current traffic address replacement.

VSITE manages VM live migration thanks to the MAC learning mechanism and by using a location IP address, which is the IP of any Layer 3 switch that interconnects data center edge and core. As such, a VM migration can be considered "live" if it takes place inside one data center edge. This migration does not modify the IP address of the VM because the VM is still connected to the same Layer 3 switch and no routing updates are required. However, if the VM does migrate to another Layer 3 switch, then the location IP address is changed and the migration is no longer "live". Thus the directory service, both edge routers, and the server's hypervisor configuration must be updated.

When a VM starts or migrates in NetLord, the NetLord agent (NLA) in the hypervisor of the correspond-

ing server will have to broadcast a NLA-HERE message to report the location of the VM to the other NLAs. The NL-ARP table entries are permanent so only one message is sufficient to update the ARP-table. However if the broadcast is lost, the ARP-table does not have the correct information. Additionally, if packets for the migrated VM are already sent, then upon arrival of those packets, the server, which does not host the VM destination any more, has to reply with an unicast NLA-NOTHERE message. When receiving a NLA-NOTHERE message, the NLA will broadcast a NLA-WHERE message in order to retrieve the correct MAC address for the migrated VM. The IP address of the VM remains unchanged throughout the migration so the session remains uninterrupted. In this way, NetLord can do live VM migration.

VNT is based on the TRILL protocol which routes the messages thanks to Layer 2 nicknames. A VM is associated to a RBridge nickname in the core network. A message for a VM is modified by the ingress RBridge which routes the frame to the egress RBridge associated with the destination VM. When a VM migrates in TRILL the only modification is the association between an RBridge nickname and the VM, except if the VM remains in the domain managed by the RBridge. As such, the IP address and MAC address of a VM is not used for routing or forwarding purposes, so they remain unchanged during a VM migration thereby preserving the session continuity and realizing a live migration. Huawei, in [93] even qualify the migration of VM with TRILL as "Smooth VM migration".

VL2, like TRILL, uses an addressing scheme which separates the server address and addresses used for routing purposes. The server addresses are called application-specific addresses (AAs) and are not modified when a VM migrates. The modified address is the location-specific address (LA) which is the one used for routing the packets. Each AA address is associated with a LA address and it is the VL2 directory system which manages those associations. When a VM migrates and changes its AA/LA association, it is the directory system which must update the mapping and thus must inform the other VMs which want to communicate with the migrated VM. As during the migration process, if neither the IP nor the MAC address of the VM changes, then the session was uninterrupted and the migration was performed live.

DOVE works with tunneling protocols such as STT, VXLAN, and NVGRE. These protocols decouple the logic domain from the physical infrastructure while respecting machine mobility, address space isolation, and multitenancy. However, to handle VM migration and address resolution for these migrated VMs, a dSwitch must, upon detection of a newly-hosted VM, send a location update to the DPS. This allows the DPS to update its address resolution information.

7.4 Resilience comparison

In this section we look at techniques such as redundancy, multipath and backup that the solutions provide in order to manage failures.

LISP resilience is done through redundancy of its components. More than one CE (Customer Edge) router with LISP capabilities can be used which translates to more than one xTR with the the same IP address. Thus the RLOC becomes an anycast address and if one of the xTR fails then the traffic is automatically routed to the other with the same address. To manage these redundant xTRs, we have two arguments. First is priority; the higher the priority, the less favorable. Second is weight; if two xTR share the same priority then the traffic is divided according to their weight. For example, if xTR1 has a weight of 10 and xTR2 a weight of 5, then the traffic ratio will be 2:1 with xTR1 receiving the double of traffic than xTR2. Additionally, Mapping Server (MS) and Mapping Resolver (MR) are key components. To assure resilience, backup devices may be needed. When using multi-homed sites, with multiple xTR, it is no longer possible for the site to control its point of entry when the anycast route is advertised. The scope of advertisement is also reduced to /32 (/128 in IPv6) prefixes.

NVGRE is an encapsulating and tunneling protocol. There is no resilience in NVGRE because the sole function of NVGRE, encapsulation, is done by an important element, the hypervisor. In the event of failure of this element the packet could not reach the destination because it could not pass the hypervisor. It is possible to add resilience on the path by using multipath techniques such as ECMP (Equal-Cost Multipath) or [94] but this is not included in NVGRE.

STT's use of a TCP-like packet but lacking all the TCP functionalities results in the loss of IP datagrams in the event of congestion, or when a router on the path is not STT-enabled. In this case the router will drop the packet. In order to prevent such an undetected packet loss, the solution is to use a real TCP header in the outer IP header. As with NVGRE, it is possible to use ECMP in addition to STT for better path resilience. However there is a necessity that all packets of the same flow follow the same path and that all paths are used efficiently. STT endpoints are designed to be virtual switches running in software. These endpoints are mostly inside the servers and so each server is an endpoint. There is no need for endpoint redundancy since if the endpoint is down then the server is down, and so are the VMs.

The resilience in 802.1ad and 802.1ah can be obtained by aggregating links. If one link is down connectivity is maintained by the remaining backup links. In a similar way redundant switches bring resilience to the network. Therefor in 802.1ad and 802.1ah resilience is accomplished by network hardware redundancy.

Like NVGRE, VXLAN is a tunneling technique. VXLAN endpoints, the VTEP, are located within the

hypervisors of each server hosting VMs. VXLAN does not define resilience techniques. Hypervisor failure is managed through the use of redundant servers and the migration of VMs to those backup servers. However the session continuity might be interrupted. Session continuity could be maintained if a faulty hypervisor is detected prior to total failure thereby allowing a backup server to join this VNI permitting live VM migration. This solution is not defined in [7] and is only a possible solution to enhance VXLAN resilience.

Since Diverter uses a Layer 2 core network it is possible to use ECMP in order to increase resiliency of the path. Additionally, a Farm's virtual gateway is distributed in all the VMs of this Farm so there is no risk that the failure of the virtual gateway will block all communications with this Farm's VMs. However if a server is down then the Farm must be replicated on another server. As there is no live migration in Diverter, all the connections must be re-established.

PortLand's core network is based on a multi-rooted fat-tree topology, which increases link capacity at the tree summit, and uses the ECMP protocol. Additionally there is redundancy at the aggregation and core level switches. However the most important element in the PortLand solution is the fabric manager. If the fabric manager fails then address resolution is no longer possible. For this reason the fabric manager should be replicated. These backups however don't need to be exact replicas, since the fabric manager does not maintain a hard state.

Sec2 uses multiple FEs per site in order to increase reliability. Also the Central Controller (CC) can have a backup if needed. Additionally the CC can become a Distributed Controller (DC). As client networks are managed independently from each other, it is possible to have several controllers, each managing different client networks. However this solution increases the administration complexity and may result in having a backup for each small controller.

BlueShield has a centralized controller, the directory server, whose role is to resolve addresses and to enforce isolations rules. If this device fails then there will be no communication between VMs as they will be unable to obtain each other's MAC address. To prevent such a situation, the directory server is replicated over several devices. Also, to improve reliability, a BlueShield agent will send its queries to several directory server devices in order to have at least one answer. If a BlueShield agent, being located in each VM, fails then logically the VM itself will have failed. The same is true for the vSwitches and ebtables, as they are located in each server. BlueShield imposes a Layer 2 network but nothing else so it is possible to use ECMP in this Layer 2 network.

There is no specific technique for resilience defined in VSITE. However a server can be multi-homed to multiple top-of-rack switches. In this case there must be a master switch to handle the locIP. This master or slave configuration of the switches is done with the virtual router redundancy protocol [95]. As VSITE is a solution with a centralized controller (the Directory server) it might be necessary to replicate this controller.

In order to benefit from a high-bandwidth resilient multipath fabric using Ethernet switches, NetLord relies on SPAIN [62]. Like the other solutions using a centralized controller, it might be necessary to have redundant configuration repositories, not only for availability but also for improvement in performance. The NetLord's agents are all located inside the hypervisors of each physical server, so for NLA redundancy we must have server redundancy.

VNT, being a distributed solution, has no need for redundant centralized controller. Additionally, VNT uses ECMP for multiple paths, so even if an RBridge fails, the traffic is sent to another RBridge. Each VNI (a.k.a tenant) can have its own multicast distribution tree and it is possible to configure a backup tree if needed.

The Clos topology, used by VL2, provides a simple and resilient topology. Routing in such a topology is done by taking a random path up to a random intermediate switch and then taking another random path down to a destination ToR switch. However VL2 has a centralized controller (the directory server) which must be replicated. Otherwise, in case of failure, address resolution would be impossible.

The use of tunneling protocols in DOVE provides multipath capabilities and routing resiliency. The key component of DOVE, the DOVE Policy Service, must be resilient to ensure high availability. It maintains the information of the network in order to resolve dSwitch policy requests. DPS should additionally be replicated and have multiple backups.

7.5 Scalability comparison

In virtualized data center and virtualized environment in general, the goal is to share the infrastructure among the maximum number of clients, thus scalability is an important criteria.

As mentioned in [96], the separation of the Endpoint Identifiers (EIDs) and Routing Locators (RLOCs) in LISP, allows for a better scalability through a greater aggregation of RLOCs. However new limits are imposed, notably one RLOC address having $2^{32} = 4294967296$ possible EIDs in IPv4 and $2^{128} \approx 3,4*10^{38}$ in IPv6. These limits are also applicable for the number of possible RLOCs, so we can see that the address space is scalable. However the MS and MR are hardware components with memory and CPU limitations. MS and MR must store mapping information between RLOCs and EIDs, but seeing as there are, for one RLOC with an IPv4 addressing, approximately 4 billion EID addresses, we can conclude that the scalability issue lies within the MS and MR components. For example in [97] the maximum number of NAT translations stored is 2147483647 which is only half of the number of EIDs in IPv4. This limitation is based on

a theoretical maximum within the Cisco IOS XE operating system, and not upon any physical hardware limitation.

The use of NVGRE endpoints allows the representation of multiple Customer Address (CA) by only one Provider Address (PA). This way the core network routers have fewer addresses to store and manage. Also the sizes of MAC address tables at the Top of Rack switches are reduced. It is also possible to increase scalability by implementing proxy ARP at each NVGRE endpoints to prevent most broadcast traffic and convert the rest to multicast traffic which reduces the load on the control plane. To prevent most broadcast traffic, NVGRE endpoints must be placed within the hypervisor. The VSID field, in the NVGRE header, is 24 bits long so there are $2^{24} = 16777216$ virtual Layer 2 networks.

In STT the core network only knows the IP addresses of each virtual switch, one virtual switch per server at most. In the worst case scenario, there is only one VM per server so there is the same number of virtual switches as VMs and since it uses an IP based address scheme, it has a similar scalability as IP including the ability to aggregate. However, usually a server hosts more than one VM so we have greater scalability. Additionally, with a Context ID of 64 bits it is possible to have $2^{64} \approx 1.8{*}10^{19}$ IDs. Consequently, scalability limitations reside in the virtual switches, which are unable to manage so many IDs. Another limit is the number of VMs per server. This last limit can be mitigated if we change the STT endpoint location. By using a dedicated device in front of several servers, the number of VMs managed by this device will be higher than if the STT endpoint was in the server itself. However, this implies additional network hardware.

Both 802.1ad and 802.1ah standards are evolutions of the 802.1Q standard with the VLAN solution. As such, they both increase the limit of VLAN from 4096 to $2^{12} * 2^{12} = 16777216$ for 802.1ad and to $2^{12} * 2^{20} = 4294967296$ for 802.1ah, and with the optional fields to $2^{12} * 2^{20} * 2^{12} * 2^{12} \approx 7 * 10^{16}$. The issue here lies with the switches, which have to manage this number of VLANs. A switch is incapable of managing so many VLANs, so in order to reduce this number, join and leave messages ensure that the switch manages only the VLANs needed by the endpoints.

VXLAN uses a VXLAN Network Identifier (VNI) which is 24 bits long so there are $2^{24} = 16777216$ VNIs. However VXLAN works at the software level, which impacts overall performance because hardware offload is not possible. Whether or not this is important, draft [98] attempted to address this question. They observed increased CPU and unstable throughput 5.6 Gb across a 10Gb network. However those results must be taken with a grain of salt,because, as stated in the paper, the tests were realized using only one server when the design and purpose of VXLAN is for a multi-server environment.

With its virtual IP addressing scheme, Diverter manages up to 16 million VMs system-wide. However the solution for IP addressing is more restrictive with regards to the number of tenants. With an IP of the type *10.F.S.H* we have 255 distinct farms, with 255 subnets in each farm. If we consider that each client uses one Sub-network then we have a maximum of $255 \times 255 = 65025$ clients in total. This limit can be modified, as stated in [8] since this address scheme is by default and may be modified prior to network deployment. With this in mind, we are faced with another scalability issue; to determine in advance how many farms, subnets, and hosts would be required. On the other hand, this addressing technique allows the core switches to only see one MAC address per server, which reduces the size of the MAC forwarding table.

As discussed in Section 5.2.1.4, PortLand is a solution with a centralized control plane. The fabric manager, a single machine, must manage all ARP traffic for the whole network, rendering this architecture unscalable in a large data center. For example in a data center with 27.648 end hosts (not tenants), each making 25 ARP requests per second, the fabric manager would need approximatively 15 CPU cores working full time just to handle the ARP requests. Additionally there is no notion of tenant isolation in the solution. In order to provide isolation, rules could be enforced by the fabric manager. For example, the fabric manager will only respond to ARP queries when allowed by policies. However, doing so increases overhead on the fabric manager, thereby further decreasing the number of end hosts that can be managed. A possible solution would be to have additional fabric manager for fewer end hosts thereby reducing the load on the fabric manager, but in [9] this solution is preceded by "it should be possible", so additional fabric manager configuration may be necessary.

Like PortLand, SEC2 is a solution using a centralized control plane. The Centralized Controller is the key element for addresses resolution, rules and isolation enforcement. The results of the CC can be extrapolated from PortLand fabric manager results. In fact those results might be "worse" seeing that the CC has more actions to do when processing an ARP request than the fabric manager. In order to reduce the load, the SEC2 CC can become a Distributed Controller with each device managing some client networks. This way we can increase the number of edge domains in the data center. Another limitation is the number of client networks in each edge domain. As the tenant isolation in edge domains is done thanks to VLANs, the number of 4096 tenants is the limit. However, the number of tenants within the DC is limited by the number of domain edges multiplied by the number of VLANs per edge domain. The limit of edge domains is the maximum number of MAC addresses. As long as there are free MAC addresses we can add edge domains. One Edge domain is associated with all its FEs' MAC address.

BlueShield improves its scalability by suppressing all

VMs' ARP broadcast and converting them to unicast directory server lookups. However the PortLand experience can be used as reference for this solution. We saw that for $\approx$ 27000 end hosts each sending 25 ARP requests per second, the centralized controller will need approximatively 15 CPU cores working non-stop to manage these requests. In order to overcome this limitation, BlueShield uses redundant directory servers to share the load. Nevertheless, contrary to SEC2, in BlueShield each directory server must have the same information. Even if we increase the number of directory servers in order to alleviate the CPU load, we will have another limitation imposed by the physical memory of the device. Additionally, the directory server must not only save the ARP information but also the rules indicating which VMs can exchange data. As a consequence of this quantity of information the DS must look through, the latency is increased.

Using locIP based on the Layer 3 switch virtual IP, VSITE can aggregate multiple VMs under one IP address. The VMs MAC addresses are only known inside the data center edges. This allows for smaller table size in core network routers as they only learn the locIP addresses. However, like the other solutions using a centralized control plane, one scalability limit is given by the capacities of the directory server which must store both IP addresses, the locIP and the real IP, MAC addresses, VLANs, and must resolve address queries. Additionally, VSITE uses VLANs for client isolation and therefore imposes a limit of 4096 virtual networks.

Concerning scalability, NetLord uses an IP encoding which gives 24 bits for the Tenant_ID value. With 24 bits it is possible to have $2^{24} = 16777216$ simultaneous tenants. The encapsulation scheme prevents Layer 2 switches to see and save all Layer 2 addresses. These Layer 2 switches see the local Layer 2 addresses and the addresses of all the edge switches. The authors of [13] estimate that NetLord can support: $N = V \times R \times \sqrt{(\frac{F}{2})}$ virtual machines. Where V is the number of VMs per physical server, R is the switch radix, and F is the MAC forwarding information base (FIB) size (in entries). In Table 1, they presented results for V = 50. Additionally NetLord use multipath technology based on SPAIN and so achieves a throughput similar to that of machine-to-machine communication.

Table 1: NetLord worst-case limits on unique MAC addresses (From [13])

Switch Radix	FIB Sizes			
	16K	32K	64K	128K
24	108,600	153,600	217,200	307,200
48	217,200	307,200	434,400	614,400
72	325,800	460,800	651,2600	921,600
94	425,350	601,600	850,700	1,203,200
120	543,000	768,000	1,086,000	1,536,000
144	651,600	921,600	1,303,200	1,843,200

The VNT solution based on TRILL has the same scalability advantages as TRILL. The core RBridges only learn the nicknames of the other RBridges. An edge RBridge aggregate multiple VMs MAC addresses under its nickname. So RBridge forwarding database sizes are reduced compared to classical Ethernet forwarding databases. [69] states :

> "... unicast forwarding tables of transit RBridges to be sized with the number of RBridges rather than the total number of end nodes ..."

It is also true if VNT is used with MLTP. Additionally, VNT introduces a VNI TAG to separate the virtual networks. This TAG is 24 bits long so it can accommodate $2^{24} = 16777216$ virtual networks which should be sufficient for the next few years.

The scalability of the VL2 solution is limited by the capacity of the directory server. In order to increase scalability, VL2 uses additional directory servers. These additional directory servers improve the maximum lookup rate. Some experimental results are given in [15]. In those experiments, the goal was to process the most lookup requests possible, while ensuring sub-10ms latency for 99% of the requests. They found that a directory server can manage 17000 lookups/sec and that the lookup rates increase linearly with the increase of servers. In the worst case scenario, chosen in [15], 100000 servers simultaneously performing 10 lookup requests requires 60 servers in the directory system. We can conclude that the scalability limitation of VL2 comes from its directory system.

DOVE's scalability is achieved thanks to the tunneling protocol it uses. As such the choice of the tunneling protocol is bound by several attributes. Among them are the interoperability and scalability attributes. Those attributes define that the protocol must used genuine headers for delivery and it must adapt to different underlays. However the scalability issue is located in the DPS. As DOVE is a solution with a centralized control plane, we have the same issue of having the centralized controller being overloaded by the amount of policy requests. So the DPS must be scalable but the means to achieve this are not specified in the article.

7.6 Multi data center comparison

Multi data center interconnection is interesting since there often may be multiple physical facilities for a given virtual data center. For this reason we will identify if the proposed solutions have inherent multi data center capabilities.

LISP is adapted for multiple data centers as long as each data center is a LISP site with at least one xTR and a RLOC address. In this case then all devices in the data center have EID addresses that are associated to the RLOC address of the xTR of the data center. In fact, [99] examines the best possible deployment of LISP in a data center and section 5 discusses data center interconnection over a wan network.

However, if some data centers are not LISP-enabled then we need to refer to RFC 6832 [100], which describes how an interconnection between a LISP site and a non LISP site is possible and implemented. This standard introduced three such mechanisms. One uses a new network element, a LISP Proxy Ingress Tunnel Router (Proxy-ITR), installed at non LISP site. Another mechanism adds another layer of Network Address Translation (NAT) at xTR. And the last also uses a new network element, a Proxy Egress Tunnel Router (Proxy-ETR).

NVGRE can be used like a site-to-site VPN. To do so each site needs a VPN gateway which supports NVGRE. These gateways will then establish a tunnel between them and encapsulate and respectively decapsulate the sent and received packets.

As mentioned in [4], "STT deployments are almost entirely limited at present to intra-data center environments". This is explained by the fact that STT uses a TCP-like header that has the same fields as a TCP header but not the same functionalities. As such, the middle boxes which do not have STT knowledge will drop the packets. That is why, for now, STT is only used in environments where the same administrative entity can manage all the middle boxes to process STT packets. So for now, even if theoretically STT can be used like a site-to-site VPN, it is not practically feasible.

802.1ad and 802.1ah can interconnect data centers if the network between them is a Layer 2 network. Most of the time however, it is a Layer 3 network thus the frames need to be encapsulated in IP. All the switches in the data center and in the Layer 2 network must respect the 802.1ad or 802.1ah standards. As both are standards, all the recent switches from manufacturers support them.

VXLAN is designed mostly for intra data center communication, however it is possible to use it like a site-to-site VPN with VXLAN gateways at each site. Additionally, [101] proposes the use of Ethernet VPN (E-VPN) technology to interconnect VXLAN sites. It could also be used to interconnect NVGRE sites. However this solution imposes the use of IP/MPLS networks between the sites.

Diverter does not specify any inter data center communication techniques. Nevertheless, each farm hosts multiple subnets, each with multiple hosts, and we can extrapolate in saying that a farm could represent a data center. This way we see that to have multiple data center interconnected with Diverter the only requirement would be to have a Layer 2 connection between those data centers. However, doing so would result in poor scalability. Additionally all the control traffic would have to reach all the VNETs from all data center which might be a costly use of the interconnection links.

PortLand is based on a fat tree topology which is a data center topology. So in order to use PortLand for inter data center communication it is mandatory to have a Layer 2 fat tree topology between the data centers. If the interconnection of each data center core switches via a Layer 2 fat-tree topology is achievable then we could achieve a large-scale PortLand network spanning multiple data centers. This being said, multi data center connectivity is not discussed in the solution brief.

By design, SEC2 is already multi-domain. We have a core domain which interconnects several edge domains. We could see the edge domains as data centers and the core domain as a Layer 2 interconnection between the data centers. Additionally we need a centralized controller reachable by all forwarding elements (FE) in all data centers. If we use a distributed controller then each member device must also be reachable by the FEs. The only issue is scalability. As tenant isolation in edge domains is done via VLANs, it means that in each data center we will have at most 4096 VLANs which is insufficient for virtualized data centers.

The BlueShield solution is based upon preventing address resolution by blocking ARP queries and converting them to unicast directory server lookups via the vSwitch. There is no notion of multi data center in the paper, however we can imagine a simple solution with a directory server replicated in each data center which manages all the rules for inter data center communication throughout the network.

By design, VSITE interconnects multiple client sites to a data center. So in the same manner we can also interconnect data centers. In order to manage this, it is necessary to increase the directory server capabilities to match the increase in information it stores. This directory server will have to store the information concerning all VMs in the network. Also each data center will need at least one cloud data center in order to implement OTV-like protocol, which exchanges MAC reachability information with other cloud data centers and the directory server.

NetLord does not address the multi data center issue. However it is possible to interconnect multiple data centers and as a result implement a larger network. All these data centers will share the same control plane, meaning that all control messages will travel across the public interconnection to reach all the data centers. This implies that the configuration repository will have to store the information for all data centers, which presents a potential scalability problem. Additionally, this interconnection will have to be done with a Layer 2 message transporting tunnel.

The TRILL protocol is multi-data-center-ready, and thereby also is VNT. However, to manage this multi data center network, the solution used by TRILL is to have one big network with one control plane shared among the data centers. This is not scalable seeing that there are only 16 bits for a nickname, 65536 nicknames in total, and that they all must be unique. This also means that the interconnection of those TRILL data centers must be done using site-to-site tunnels. However, when using the MLTP/VNT solution, the

merging issue does not exist anymore as each data center control plane remains independent. Additionally, MLTP introduce a new nickname management which increase the number of available nicknames to more than one billion. Nevertheless, even when using MLTP, the interconnection of MLTP data centers must be done using site-to-site tunnels.

The multi data center issue is not discussed in VL2. It might however be possible to interconnect multiple data centers. The directory system could be externalized in order to manage the whole network, which could span multiple data centers.

Like VL2, DOVE does not address the multi data centers issue. Nevertheless it might be possible to have the DOVE solution span over multiple data centers. However, this means that the DPS will have to manage the request of even more dSwitches from all the data centers. This way the control plane is shared among all the data centers and the DPS redundancy is the new scalability limit.

8 Discussion

Tunneling solutions based on a centralized controller need to tackle the scalability issue. Current solutions with a centralized control plane need a centralized controller with substantial processing power or such devices are costly. A solution could be to use multiple devices aggregated to form a centralized controller in order to share the load. However, those devices have to be synchronized. Another solution could be to have smaller interconnected data center. However the interconnection between data centers is not really addressed. Even if some architectures are multi-site by design, those have scalability issues inside each site which prevents the site from being a data center. For those architectures, the solution might be found in using another tunneling protocol enabling a better scalability within a site.

The next possible extension of cloud computing is the hybrid cloud. Gartner [102] expects that hybrid cloud will be adopted by almost half of the largest companies by the end of 2017. In this type of cloud, tenants' traffic will be indistinctly crossing the public network between the data center network and a tenant's private network. And as in public cloud, tenants would want their traffic to be isolated from other tenants and other entities in general. The presented solutions improve tenant data security thanks to traffic isolation achieved by respecting rules and forwarding tables. However, isolation is only a part of security. Security is an area to improve as isolation is not sufficient to guaranty integrity and prevent the theft of the data. For example, if a corrupt or faulty component does not respect these rules then tenants' traffic isolation is compromised. Another case could be that a malicious user or even a data center employee might illegally access a central component. It could allow them to arbitrarily implement any rules they desire and thereby override isolation rules, even if network components correctly adhere to them. An attacker could also realize a man-in-the-middle attack or intercept the traffic, or as data centers are now more and more virtualized, with VMs migrating across these data centers to improve performance, or in case of necessity, there are more and more tenants and their data is passing through more and more devices increasing the risk for the data. Additionally, now that hybrid cloud is growing, work must be done in order to isolate traffic from end-to-end between the tenant's private network and the cloud's infrastructure.

Any solution must also take into account that the security requirements of one tenant may be considerably different to that of another. This demonstrates a need to manage several security mechanisms and policies, which increases the complexity of the network. In addition, there are potential conflicts between intrusion detection systems policies, belonging to service or infrastructure providers, and firewalls, which need to be resolved [103].

Another topic to tackle is the fact that Layer 2 solutions mostly use Spanning Tree (STP), Rapid Spanning Tree (RSTP), or Multiple Spanning Tree (MSTP) to prevent loops, thus rendering unusable a number of links and reducing the overall performance of the data center. To prevent that, a solution could be to use level 2 multipath technology. TRILL possesses such functionality and other Layer 2 solutions could use Shortest Path Bridging (SPB) specified in the IEEE 802.1aq standard.

Another area of improvement for Layer 2 solution is CPU offloading. Some Layer 2 solutions add a new header which is not yet recognized by Network Interface Cards (NICs) thus the processing of the packet is done by the CPU which consumes additional resources and decreases overall performance. A practical solution could be to program the offloading of these new headers in NICs or to distribute traffic across multiple CPUs.

9 Conclusion

Data centers are being more frequently used. Especially cloud data centers where workloads, representing 39% of total data center workloads, will continue to grow, up to 63% of the total data center workloads by 2017. The cloud data center advantage is that it hosts multiple tenants to increase infrastructure efficiency and reduce costs. However some issues arose like tenants' traffic isolation.

In this paper, we surveyed fifteen solutions that provide tenant traffic isolation in a cloud network. We first presented them and then compared their complexity, the overhead they induce, their abilities to manage VMs migration, their resilience, their scalability, and their multi data center capabilities. Each solution provides tenant traffic isolation by using varying approaches, however these solutions are not all multi-data-center-ready, and those that are have potential

issues with scalability. Nevertheless, VNT solution based on TRILL derives multi data center capability from the work already done on trill, implementing control plane isolation in each data center and an interconnection network control plane, thereby increasing scalability [104, 105, 106, 72].

Finally we identified some research areas which are not yet thoroughly discussed in these papers, and are areas for possible future research. Tenant traffic is not safe enough by just isolating it. It may be necessary to implement other security mechanisms in order to provide better security.

Data centers are increasingly being virtualized, with VMs migrating across these data centers to improve performance or in case of necessity. However the interconnection between data centers is not really addressed. When a multi data center technique is presented, there is a trade off in the scalability of the solution.

Additionally, now that hybrid clouds are growing, work must be done in order to isolate traffic from end to end between a tenant's private network and the cloud.

References

[1] Cisco global cloud index: Forecast and methodology, 2013-2018. Technical report, Cisco Systems, Inc, 2014. http://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/Cloud_Index_White_Paper.html.

[2] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis. The locator/id separation protocol (lisp). RFC 6830, January 2013.

[3] Murari Sridharan, Yu-Shun Wang, Albert Greenberg, Pankaj Garg, Narasimhan Venkataramiah, Kenneth Duda, Ilango Ganga, Geng Lin, Mark Pearson, Patricia Thaler, and Chait Tumuluri. Nvgre: Network virtualization using generic routing encapsulation. Work in progress, draft-sridharan-virtualization-nvgre-04, February 2014.

[4] Bruce Davie and Jesse Gross. A stateless transport tunneling protocol for network virtualization (stt). Work in progress, draft-davie-stt-06, April 2014.

[5] Institute of Electrical and Electronics Engineers. Ieee 802.1ad-2005. 802.1ad - Virtual Bridged Local Area Networks, 2005.

[6] Institute of Electrical and Electronics Engineers. Ieee 802.1ah-2008. 802.1ah - Provider Backbone Bridges, 2008.

[7] Mallik Mahalingam, Dinesh G. Dutt, Kenneth Duda, Puneet Agarwal, Lawrence Kreeger, T. Sridhar, Mike Bursell, and Chris Wright. Vxlan: A framework for overlaying virtualized layer 2 networks over layer 3 networks. Work in progress, draft-mahalingam-dutt-dcops-vxlan-09, April 2014.

[8] Aled Edwards, Anna Fischer, and Antonio Lain. Diverter: A new approach to networking within virtualized infrastructures. In *Proceedings of the 1st ACM Workshop on Research on Enterprise Networking*, WREN '09, pages 103–110, New York, NY, USA, 2009. ACM. http://doi.acm.org/10.1145/1592681.1592698.

[9] Radhika Niranjan Mysore, Andreas Pamboris, Nathan Farrington, Nelson Huang, Pardis Miri, Sivasankar Radhakrishnan, Vikram Subramanya, and Amin Vahdat. Portland: A scalable fault-tolerant layer 2 data center network fabric. In *Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication*, SIGCOMM '09, pages 39–50, New York, NY, USA, 2009. ACM. http://doi.acm.org/10.1145/1592568.1592575.

[10] Fang Hao, T. V. Lakshman, Sarit Mukherjee, and Haoyu Song. Secure cloud computing with a virtualized network infrastructure. In *Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing*, HotCloud'10, pages 16–16, Berkeley, CA, USA, 2010. USENIX Association. http://dl.acm.org/citation.cfm?id=1863103.1863119.

[11] Saurabh Barjatiya and Prasad Saripalli. Blueshield: A layer 2 appliance for enhanced isolation and security hardening among multi-tenant cloud workloads. In *Proceedings of the 2012 IEEE/ACM Fifth International Conference on Utility and Cloud Computing*, UCC '12, pages 195–198, Washington, DC, USA, 2012. IEEE Computer Society. http://dx.doi.org/10.1109/UCC.2012.21.

[12] Li Li and Thomas Woo. Vsite: A scalable and secure architecture for seamless l2 enterprise extension in the cloud. In *Secure Network Protocols (NPSec), 2010 6th IEEE Workshop on*, pages 31–36. IEEE, 2010.

[13] Jayaram Mudigonda, Praveen Yalagandula, Jeff Mogul, Bryan Stiekes, and Yanick Pouffary. Netlord: A scalable multi-tenant network architecture for virtualized datacenters. In *Proceedings of the ACM SIGCOMM 2011 Conference*, SIGCOMM '11, pages 62–73, New York, NY, USA, 2011. ACM. http://doi.acm.org/10.1145/2018436.2018444.

[14] Ahmed Amamou, Kamel Haddadou, and Guy Pujolle. A trill-based multi-tenant data center network. *Computer Networks*, 68(0):35 – 53, 2014. Communications and Networking in the Cloud http://www.sciencedirect.com/science/article/pii/S1389128614000851.

[15] Albert Greenberg, James R. Hamilton, Navendu Jain, Srikanth Kandula, Changhoon Kim, Parantap Lahiri, David A. Maltz, Parveen Patel, and Sudipta Sengupta. Vl2: A scalable and flexible data center network. In *Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication*, SIGCOMM '09, pages 51–62, New York, NY, USA, 2009. ACM. http://doi.acm.org/10.1145/1592568.1592576.

[16] Liane Lewin-Eytan, Katherine Barabash, Rami Cohen, Vinit Jain, and Anna Levin. Designing modular overlay solutions for network virtualization. Technical report, IBM, 2011.

[17] R. Cohen, K. Barabash, V. Jain, R. Recio, and B. Rochwerger. Dove: Distributed overlay virtual network architecture, 2012.

[18] Rouven Krebs, Christof Momm, and Samuel Kounev. Architectural concerns in multi-tenant saas applications. In *CLOSER*, pages 426–431, 2012.

[19] Cisco virtualized multi-tenant data center, version 2.0 compact pod design guide. Technical report, Cisco Systems, Inc, 2010. http://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Data_Center/VMDC/2-0/design_guide/vmdcDesignGuideCompactPoD20.pdf.

[20] Cisco virtualized multi-tenant data center, version 2.2 design guide. Technical report, Cisco Systems, Inc, 2012. http://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Data_Center/VMDC/2-2/design_guide/vmdcDesign22.pdf.

[21] Securing multi-tenancy and cloud computing. Technical report, Juniper Networks, Inc, 2012. https://www.juniper.net/us/en/local/pdf/whitepapers/2000381-en.pdf.

[22] Steve Bobrowski. The force.com multitenant architecture. Technical report, salesforce.com, inc, 2013. http://s3.amazonaws.com/dfc-wiki/en/images/8/8b/Forcedotcom-multitenant-architecture-wp-2012-12.pdf.

[23] Virtual network overview. https://msdn.microsoft.com/en-us/library/azure/jj156007.aspx.

[24] N.M. Mosharaf Kabir Chowdhury and Raouf Boutaba. A survey of network virtualization. *Computer Networks*, 54(5):862 – 876, 2010.

[25] A. Fischer, J.F. Botero, M. Till Beck, H. de Meer, and X. Hesselbach. Virtual network embedding: A survey. *Communications Surveys Tutorials, IEEE*, 15(4):1888–1906, Fourth 2013.

[26] Tunneling - cisco. http://www.cisco.com/c/en/us/products/ios-nx-os-software/tunneling/index.html.

[27] What is a tunneling protocol? http://usa.kaspersky.com/internet-security-center/definitions/tunneling-protocol.

[28] Vpn tunneling protocols. https://technet.microsoft.com/en-us/library/dd469817%28v=ws.10%29.aspx.

[29] Li heng, Yang dan, and Zhang xiaohong. Survey on multi-tenant data architecture for saas. *IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 6, No 3, November 2012*, 2012.

[30] Stefan Aulbach, Torsten Grust, Dean Jacobs, Alfons Kemper, and Jan Rittinger. Multi-tenant databases for software as a service: Schema-mapping techniques. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, pages 1195–1206, New York, NY, USA, 2008. ACM.

[31] Vivek Narasayya, Sudipto Das, Manoj Syamala, Badrish Chandramouli, and Surajit Chaudhuri. Sqlvm: Performance isolation in multi-tenant relational database-as-a-service. In *6th Biennial Conference on Innovative Data Systems Research (CIDR '13)*, 2013.

[32] Ying Hua Zhou, Qi Rong Wang, Zhi Hu Wang, and Ning Wang. Db2mmt: A massive multi-tenant database platform for cloud computing. In *e-Business Engineering (ICEBE), 2011 IEEE 8th International Conference on*, pages 335–340, Oct 2011.

[33] Xuequan Zhou, Dechen Zhan, Lanshun Nie, Fanchao Meng, and Xiaofei Xu. Suitable database development framework for business component migration in saas multi-tenant model. In *Service Sciences (ICSS), 2013 International Conference on*, pages 90–95, April 2013.

[34] Wang Xue, Li Qingzhong, and Kong Lanju. Multiple sparse tables based on pivot table for multi-tenant data storage in saas. In *Information and Automation (ICIA), 2011 IEEE International Conference on*, pages 634–637, June 2011.

[35] Ahmed Amamou. *Network isolation in a virtualized datacenter*. PhD thesis, University Pierre and Marie Curie - Paris 6 - EDITE of Paris, 2013. French thesis, Isolation reseau dans un datacenter virtualise.

[36] Jinho Hwang, Sai Zeng, F.Y. Wu, and T. Wood. A component-based performance comparison of four hypervisors. In *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on*, pages 269–276, May 2013.

[37] Hasan Fayyad-Kazan, Luc Perneel, and Martin Timmerman. Benchmarking the performance of microsoft hyper-v server, vmware esxi and xen hypervisors. *Journal of Emerging Trends in Computing and Information Sciences*, 4(12), 2013.

[38] Todd Deshane, Zachary Shepherd, J Matthews, Muli Ben-Yehuda, Amit Shah, and Balaji Rao. Quantitative comparison of xen and kvm. *Xen Summit, Boston, MA, USA*, pages 1–2, 2008.

[39] Wei Jing, Nan Guan, and Wang Yi. Performance isolation for real-time systems with xen hypervisor on multi-cores. In *Embedded and Real-Time Computing Systems and Applications (RTCSA), 2014 IEEE 20th International Conference on*, pages 1–7, Aug 2014.

[40] Stan Hanks, Tony Li, Dino Farinacci, and Paul Traina. Generic routing encapsulation (gre). RFC 1701, October 1994.

[41] Dino Farinacci, Tony Li, Stan Hanks, David Meyer, and Paul Traina. Generic routing encapsulation (gre). RFC 2784, March 2000.

[42] K. Hamzeh, G. Pall, W. Verthein, J. Taarud, W. Little, and G. Zorn. Point-to-point tunneling protocol (pptp). RFC 2637, July 1999.

[43] W. Townsley, A. Valencia, A. Rubens, G. Pall, G. Zorn, and B. Palter. Layer two tunneling protocol "l2tp". RFC 2661, August 1999.

[44] J. Lau, M. Townsley, and I. Goyret. Layer two tunneling protocol - version 3 (l2tpv3). RFC 3931, March 2005.

[45] S. Bryant and P. Pate. Pseudo wire emulation edge-to-edge (pwe3) architecture. RFC 3985, March 2005.

[46] Institute of Electrical and Electronics Engineers. Ieee 802.1q-2005. 802.1q - Virtual Bridged Local Area Networks, 2005.

[47] Institute of Electrical and Electronics Engineers. Ieee 802.1d-1990. 1990.

[48] Ieee.org, 802.1ad - provider bridges. `http://www.ieee802.org/1/pages/802.1ad.html`.

[49] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol label switching architecture. RFC 3031, January 2001.

[50] Bob Braden, Lixia Zhang, Steve Berson, Shai Herzog, and Sugih Jamin. Resource reservation protocol (rsvp) – version 1 functional specification. RFC 2205, September 1997.

[51] Loa Andersson, Ross Callon, Ram Dantu, Paul Doolan, Nancy Feldman, Andre Fredette, Eric Gray, Juha Heinanen, Bilel Jamoussi, Timothy E. Kilty, and Andrew G. Malis. Constraint-based lsp setup using ldp. RFC 3212, January 2002.

[52] Kathleen Nichols, Steven Blake, Fred Baker, and David L. Black. Definition of the differentiated services field (ds field) in the ipv4 and ipv6 headers. RFC 2474, December 1998.

[53] L. Berger. Generalized multi-protocol label switching (gmpls) signaling functional description. RFC 3471, January 2003.

[54] E. Mannie. Generalized multi-protocol label switching (gmpls) architecture. RFC 3945, October 2004.

[55] E. Rosen and Y. Rekhter. Bgp/mpls ip virtual private networks (vpns). RFC 4364, February 2006.

[56] Jon Brodkin. Vmware users pack a dozen vms on each server, despite memory constraints. `http://www.networkworld.com/article/2197837/virtualization/vmware-users-pack-a-dozen-vms-on-each-server--despite-memory-constraints.html`.

[57] Lori MacVittie. Virtual machine density as the new measure of it efficiency. `https://devcentral.f5.com/articles/virtual-machine-density-as-the-new-measure-of-it-efficiency`.

[58] Determine true total cost of ownership. `http://www.vmware.com/why-choose-vmware/total-cost/virtual-machine-density.html`.

[59] Rich Miller. A look inside amazon's data centers. `http://www.datacenterknowledge.com/archives/2011/06/09/a-look-inside-amazons-data-centers/`.

[60] Rich Miller. Who has the most web servers? `http://www.datacenterknowledge.com/archives/2009/05/14/whos-got-the-most-web-servers/`.

[61] Kireeti Kompella. New take on sdn: Does mpls make sense in cloud data centers? `http://www.sdncentral.com/use-cases/does-mpls-make-sense-in-cloud-data-centers/2012/12/`.

[62] Jayaram Mudigonda, Praveen Yalagandula, Mohammad Al-Fares, and Jeffrey C. Mogul. Spain: Cots data-center ethernet for multipathing over arbitrary topologies. In *Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation*, NSDI'10, pages 18–18, Berkeley, CA, USA, 2010. USENIX Association. `http://dl.acm.org/citation.cfm?id=1855711.1855729`.

[63] Charles Clos. A study of non-blocking switching networks. *Bell System Technical Journal, The*, 32(2):406–424, March 1953.

[64] Ronald van der Pol. Ieee 802.1ah basics (provider backbone bridges), March 2011.

[65] Marco Foschiano and Sanjib HomChaudhuri. Cisco systems' private vlans: Scalable security in a multi-client environment. RFC 5517, February 2010.

[66] Danny McPherson and Barry Dykes. Vlan aggregation for efficient ip address allocation. RFC 3069, February 2001.

[67] Charles E. Leiserson. Fat-trees: Universal networks for hardware-efficient supercomputing. *IEEE Trans. Comput.*, 34(10):892–901, October 1985. `http://dl.acm.org/citation.cfm?id=4492.4495`.

[68] Radia Perlman. Rbridges: Transparent routing. In *Proceedings of the IEEE INFOCOMM 2004*, INFOCOMM '04, 2004.

[69] Radia Perlman, Donald E. Eastlake 3rd, Dinesh G. Dutt, Silvano Gai, and Anoop Ghanwani. Routing bridges (rbridges): Base protocol specification. RFC 6325, July 2011.

[70] David R. Oran. Osi is-is intra-domain routing protocol. RFC 1142, February 1990.

[71] *Information technology – Telecommunications and information exchange between systems – Intermediate system to Intermediate system intra-domain routeing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode Network Service.* (ISO 8475), ISO/IEC 10589, 1992.

[72] Valentin Del Piccolo, Ahmed Amamou, William Dauchy, and Kamel Haddadou. Multi-tenant isolation in a trill based multi-campus network. In *Cloud Networking (CloudNet), 2015 IEEE 4th International Conference on*, pages 51–57, Oct 2015.

[73] Russell Housley and Scott Hollenbeck. Etherip: Tunneling ethernet frames in ip datagrams. *Network Working Group, Request for Comments*, 3378, 2002.

[74] Cisco. *Cisco Active Network Abstraction Reference Guide, 3.7*, June 2010. Part 2 - Technology Support and Information Model Objects : Virtual Routing and Forwarding.

[75] S Kent and K Seo. Security architecture for the internet protocol. RFC 4301, December 2005.

[76] Alan O. Freier, Philip Karlton, and Paul C. Kocher. The secure sockets layer (ssl) protocol version 3.0. RFC 6101, August 2011.

[77] Openstack security guide. Technical report, OpenStack Foundation, 2014. `http://docs.openstack.org/security-guide/security-guide.pdf`.

[78] Openflow. `https://www.opennetworking.org/sdn-resources/onf-specifications/openflow`.

[79] Amazon virtual private cloud. `http://aws.amazon.com/vpc/`.

[80] Dan Harkins and Dave Carrel. The internet key exchange (ike). RFC 2409, November 1998.

[81] Yakov Rekhter, Tony Li, and Susan Hares. A border gateway protocol 4 (bgp-4). RFC 4271, January 2006.

[82] Geoffrey Huang, Stephane Beaulieu, and Dany Rochefort. A traffic-based method of detecting dead internet key exchange (ike) peers. RFC 3706, February 2004.

[83] Pekka Savola. Mtu and fragmentation issues with in-the-network tunneling. RFC 4459, April 2006.

[84] Defense Advanced Research Projects Agency Information Processing Techniques Office. Internet protocol - darpa internet program - protocol specification. RFC 4459, September 1981.

[85] V. Fuller and D. Farinacci. Locator/id separation protocol (lisp) map-server interface. RFC 6833, January 2013.

[86] Fred Baker, Eliot Lear, and Ralph Droms. Procedures for renumbering an ipv6 network without a flag day. RFC 6325, September 2005.

[87] Charles E. Perkins. Ip mobility support for ipv4, revised. RFC 5944, November 2010.

[88] Charles E. Perkins, David B. Johnson, and Jari Arkko. Mobility support in ipv6. RFC 6275, July 2011.

[89] Jari Arkko, Christian Vogt, and Wassim Haddad. Enhanced route optimization for mobile ipv6. RFC 4866, May 2007.

[90] Dino Farinacci, Darrel Lewis, David Meyer, and Chris White. Lisp mobile node. work in progress, draft-meyer-lisp-mn-10, January 2014.

[91] Bhumip Khasnabish, Bin Liu, Baohua Lei, and Feng Wang. Mobility and interconnection of virtual machines and virtual network elements. work in progress, draft-khasnabish-vmmi-problems-03, December 2012.

[92] Murari Sridharan, Yu-Shun Wang, Pankaj Garg, and Praveen Balasubramanian. Nvgre-ext: Network virtualization using generic routing encapsulation extensions. work in progress, draft-sridharan-virtualization-nvgre-ext-02, June 2014.

[93] Technology white paper - trill. Technical report, HUAWEI TECHNOLOGIES CO., LTD., 2013. `http://www.huawei.com/ilink/enenterprise/download/HW_259594`.

[94] Alan Ford, Costin Raiciu, Mark Handley, and Olivier Bonaventure. Tcp extensions for multipath operation with multiple addresses. RFC 6824, January 2013.

[95] Stephen Nadas. Virtual router redundancy protocol (vrrp) version 3 for ipv4 and ipv6. RFC 5798, March 2010.

[96] Lisp overview... `http://lisp.cisco.com/lisp_over.html`.

[97] Ip addressing: Nat configuration guide, cisco ios xe release 3s. `http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipaddr_nat/configuration/xe-3s/nat-xe-3s-book.pdf`.

[98] Vic Liu, Bob Mandeville, Brooks Hickman, Weiguo Hao, and Zu Qiang. Problem statement for vxlan performance test. work in progress, draft-liu-nvo3-ps-vxlan-perfomance-00, July 2014.

[99] Victor Moreno, Fabio Maino, Darrel Lewis, Michael Smith, and Satyam Sinha. Lisp deployment considerations in data center networks. work in progress, draft-moreno-lisp-datacenter-deployment-00, February 2014.

[100] Darrel Lewis, David Meyer, Dino Farinacci, and Vince Fuller. Interworking between locator/id separation protocol (lisp) and non-lisp sites. RFC 6832, January 2013.

[101] Sami Boutros, Ali Sajassi, Samer Salam, Dennis Cai, Samir Thoria, Tapraj Singh, John Drake, and Jeff Tantsura. Vxlan dci using evpn. work in progress, draft-boutros-l2vpn-vxlan-evpn-04, July 2014.

[102] Gartner says nearly half of large enterprises will have hybrid cloud deployments by the end of 2017. `https://www.gartner.com/newsroom/id/2599315`.

[103] E. Al-Shaer, H. Hamed, R. Boutaba, and M. Hasan. Conflict classification and analysis of distributed firewall policies. *IEEE Journal on Selected Areas in Communications*, 23(10):2069–2084, 2005.

[104] Radia Perlman, Donald Eastlake, Anoop Ghanwani, and Hongjun Zhai. Flexible multilevel trill (transparent interconnection of lots of links). Work in progress, draft-perlman-trill-rbridge-multilevel-07, January 2014.

[105] Sam Aldrin, Donald Eastlake, Tissa Senevirathne, Ayan Banerjee, and Santiago Alvarez. Trill data center interconnect. Work in progress, draft-aldrin-trill-data-center-interconnect-00, March 2012.

[106] Tissa Senevirathne, Les Ginsberg, Sam Aldrin, and Ayan Banerjee. Default nickname based approach for multilevel trill. Work in progress, draft-tissa-trill-multilevel-02, Mars 2013.

Valentin Del Piccolo is a Phd student at the University Pierre et Marie Curie (UPMC) and at GANDI SAS where he works on virtualization and multi-tenant isolation in data centers networks. He received his M.S degree in network and computer science from the University Pierre et Marie Curie in 2013, Paris, France.

Ahmed Amamou is a research engineer at GANDI SAS. He received the engineer degree in computer science from the National School of Computer science (Tunisia) in 2009 and the M.S degree in network and computer science from the same school in 2011; and the Ph.D degree in network and computer science from the University Pierre et Marie Curie in 2013, Paris, France. His research interests are Cloud computing and virtualization technologies. He is a member of the IEEE.

Kamel Haddadou received the engineering degree in computer science from INI in 2000, the M.S degree in data processing methods for industrial systems from the University of Versailles, and the PhD degree in computer networks from University Pierre et Marie Curie (UPMC), in 2002 and 2007, respectively. In 2001, he was a research assistant at the Advanced

Technology Development Centre (CDTA), Algiers, Algeria. He is currently a research fellow at the Gandi SAS, France. Since 2003, he has been involved in several projects funded by the European Commission and the French government (RAVIR, ADANETS, Adminroxy, GITAN, OGRE, ADANETS, MMQoS, SAFARI, and ARCADE). His research interests are focused primarily on Cloud computing and on resource management in wired and wireless networks. He is equally interested in designing new protocols and systems with theoretical concepts, and in providing practical implementations that are deployable in real environments. He has served as the TPC member for many international conferences, including IEEE ICC, GLOBECOM, and reviewer on a regular basis for major international journals and conferences in networking. He is a member of the IEEE.

Guy Pujolle received the PhD and "These d'Etat" degrees in computer science from the University of Paris IX and Paris XI in 1975 and 1978, respectively. He is currently a professor at University Pierre et Marie Curie (UPMC - Paris 6), a distinguished invited professor at POSTECH, Korea, and a member of the Institut Universitaire de France. During 1994-2000, he was a professor and the head of the Computer Science Department of Versailles University. He was also the professor and the head of the MASI Laboratory at Pierre et Marie Curie University (1981-1993), professor at ENST (1979-1981), and a member of the scientific staff of INRIA (1974-1979). He is the French representative at the Technical Committee on Networking at IFIP. He is an editor for ACM International Journal of Network Management, Telecommunication Systems, and an editor-in-chief of Annals of Telecommunications. He is a pioneer in high-speed networking having led the development of the first Gbit/s network to be tested in 1980. He has participated in several important patents like DPI or virtual networks. He is the cofounder of QoSMOS (www.qosmos.fr), Ucopia Communications (www.ucopia.com), Ginkgo-Networks (www.ginkgo-networks.com), EtherTrust (www.ethertrust.com), Virtuor (www.VirtuOR.fr), and Green Communications (www.greencommunications.fr). He is a senior member of the IEEE.

Table 2: Comparison of the protocols' complexities

Host isolation	**Diverter**	**BlueShield**	**NetLord**	**VL2**	**DOVE**	**LISP**	**NVGRE**	**STT**
Control plane	Distributed	Centralized, Directory Server (DS) Possible redundancy of DS	Centralized, Configuration repository	Centralized, directory system (ds)	Centralized, DOVE Policy service	Centralized, Directory Name Server (DNS)	Distributed	Distributed
Network restriction(s)	Flat Layer 2	Layer 2	Layer 2 and edge switches supporting IP forwarding	Layer 3 and Clos topologyg	Layer 3	Layer 3	Layer 3 network and No fragmentation of NVGRE packets	Layer 3 network and middle boxes (firewalls) must permit STT packets
Tunnel configuration and establishment *	Implicit	Implicit	Implicit	Implicit	Encapsulation protocol dependent	Implicit	Implicit	Implicit
Tunnel management and maintenance	Yes with forwarding table and rules in VNET	Yes, rules in the DS	Yes, with a SPAIN agent	Yes, mapping in the ds and VBL protocol	Encapsulation protocol dependent	Yes with mapping in the ITR and ETR	None	None
Multi-protocol	No, IP	Yes, Layer 2	No, Ethernet	No, IP	Encapsulation protocol dependent	No, IP	Yes, Layer-2	No, Ethernet
Security mechanism	VNET scans the traffic to enforce rules	Echelon VMs scan the traffic to enforce rules	None	VL2 agents enforce the rules	None	None	None	None

Core isolation	**PortLand**	**SEC2**	**802.1ad**	**802.1ah**	**VSITE**	**VNT**	**VXLAN**
Control plane	Centralized, Fabric manager for forwarding and addressing	Centralized, Central Controller (CC)	No	No	Centralized, Directory Server	No	No
Network restriction(s)	Layer 2 multi-rooted fat-tree	Layer 2	None	None	None	Layer 2 with TRILL enabled edges switch	Layer 3 network, no fragmentation of VXLAN packets and IGMP querier function
Tunnel configuration and establishment *	Implicit	Implicit	Explicit, GVRP	Explicit, GVRP	Explicit, MPLS VPN on public network and implicit in *vstub*	Implicit	Implicit
Tunnel management and maintenance	Yes, soft states	Yes, rules in CC	GVRP, join and leave messages by both end stations and Bridges	GVRP, join and leave messages by both end stations and Bridges	Yes, mapping in directory server and hypervisor	Yes, temporary forwarding database entry in RBridges	Join and leave messages by VTEPs
Multi-protocol	Yes	No, IP	No, Ethernet	No, Ethernet	No, Ethernet	Yes	No, Ethernet
Security mechanism	None	FEs enforce CC rules	None	None	Hypervisors enforce rules of directory server	None	None

* **Implicit**:based on connectionless IP service model. **Explicit**:tunnel establishing procedure such as control messages exchange or registration procedures.

Table 3: Comparison of the protocols' overhead

Host isolation	**Diverter**	**BlueShield**	**NetLord**	**VL2**	**DOVE**
Encapsulation header	None but IP addresse restriction	None	MAC and IP encapsulation with address rewriting (MAC+IPv4= 304 bits, MAC+IPv6= 464 bits)	IP header with a LA address (160 or 320 bits)	NVGRE, STT, or VXLAN headers
Messages	Multicast ARP messages	Directory look-up request	Adddress resolution based on Diverter model. SPAIN agent request to the repository	Messages for registration and mapping. Look-up requests. Messages for directory. IP-based link state routing protocol for LA address assignation	Policy requests Messages for registration of rules and topology.
Component(s)	VNET in each physical host with VNET ARP engine	Directory server, vSwitch, ebtables firewall, BlueShield agent, Echelon VM	NetLord Agent (NLA), SPAIN agent, Edge switches with IP routing capacities, Configuration repository	VL2 agent, Directory system	dSwitches, DOVE Policy Service (DPS)

Host isolation	**LISP**	**NVGRE**	**STT**
Encapsulation header	Outer IP header(IPV4: 160 bits, IPV6: 320 bits) + UDP header(64 bits)+ LISP header (64 bits) = 288(IPv4) or 448(IPv6) bits	Outer Ethernet header (144 bits) + Outer IP header(IPV4: 160 bits, IPV6: 320 bits) + NVGRE header (64 bits) = 368(IPv4) or 528(IPv6) bits	Outer Ethernet header (144 bits) + Outer IP header(IPV4: 160 bits, IPV6: 320 bits) + TCP-Like header(192 bits) + STT header(144 bits) = 640(IPv4) or 800(IPv6) bits
Messages	Map-Request, Map-Reply Map-Register, Map-Notify Encapsulated Control Message	None	None
Component(s)	xTR (ITR,ETR,PETR,PITR)	NVGRE Endpoints	STT Endpoints

Core isolation	**PortLand**	**SEC2**	**VSITE**	**VNT**
Encapsulation header	None	MAC header (144 bits)	MAC-in-MAC in Layer 2 network (144 bits) IP encapsulation in Layer 3 network (IPV4: 160 bits, IPV6: 320 bits)	Encapsulation with a VNT header (192 bits)
Messages	Unicast ARP in best case Worst case ARP broadcast to all end hosts Location Discovery Protocol messages Registration messages	Unicast ARP Customer messages for CC rules Uses GARP protocol	OTV-like protocol messages Directory lookup request	TRILL messages IS-IS protocol messages SPF tree generated based on Link State PDU (LSP) messages
Component(s)	Edge switches must perform MAC to PMAC header rewriting	Central Controller, Forwarding Elements, web portal	Directory server Cloud data center CEc VSITE agent	RBridges Virtual Switch

Core isolation	**802.1ad**	**802.1ah**	**VXLAN**
Encapsulation header	S-TAG (32 bits) + C-TAG (32 bits) = 64 bits	B-DA(48 bits) + B-SA(48 bits) + B-TAG(32 bits) + I-TAG(48 bits) = 176 bits +(optional) S-TAG (32 bits) + C-TAG (32 bits) = 240 bits	Outer Ethernet header(144 bits) + Outer IP Header: (IPv4=160 or IPv6=320 bits) + Outer UDP header: (64 bits) + VXLAN header: (64 bits) = 432(IPv4) or 592(IPv6)
Messages	Generic Attribute Registration Protocol	Generic Attribute Registration Protocol	Join and leave messages
Component(s)	Devices must abide by the 802.1ad standard	Devices must abide by the 802.1ah standard	VXLAN Tunnel EndPoints (VTEP)

Table 4: Comparison of the Host isolation protocols

Host isolation	**Diverter**	**BlueShield**	**NetLord**	**VL2**
Migration	Migration live or offline depending on time out values.	Live migration.	Live migration. Uses NetLord Agent messages (NLA): NLA-HERE, NLA-NOTHERE and NLA-WHERE to signal the VM migration. VM's IP or MAC address unchanged.	Live migration. Separation of location Addresses (LA) and application-specific addresses (AA).
Resilience	ECMP for multipath. Virtual gateway distributed among all the VM of the Sub-network.	Multiple replicas of the directory server. Possibility to use ECMP.	Relies on SPAIN for multipath. Configuration Repository might be replicated.	Resilience provided by a Clos topology. Redundancy of the Directory server
Scalability	16 millions VMs system wide. However number of client depend on the division of the IP address. The division must be done before starting the network, no modification after.	Centralized controller. CPU load lessen by replicating directory server but memory is limited.	16777216 Tenant_IDs (24 bits) $V \times R \times \sqrt{(\frac{F}{2})}$ virtual machines V = number of VMs per physical server R = switch radix, F = FIB size in entries.	One directory server can manage up to 17000 lookups/sec. The lookup rates increase linearly with the increase of servers.
Multi data center	Not specified. Possible with a Layer 2 interconnection. Control traffic travel between DC. Creates one big network over multiple DC	Not specified. Possibly a directory server replicated in each data center for inter-data centers communication rules.	Not specified. But possible with Layer 2 tunnels between data centers and one control plane spanning over all the data centers.	Not specified but possible. It will require an important directory system to manage the whole network.

Host isolation	**DOVE**	**LISP**	**NVGRE**	**STT**
Migration	Tunneling protocol dependent. Additionally dSwitch must inform the DPS when a new VM is detected by it.	IPv4 Mobility (RFC5944), IPv6 Mobility (RFC 6275, RFC 4866). Endpoint is an xTR itself.	REDIRECT messages	No STT mechanisms
Resilience	Multipath and routing resilience thanks to tunnel protocol. Redundancy of the Dove Policy Server.	Redundancy of xTR, MR and MS.	Multipath possible but not included in NVGRE	Multipath (ECMP) possible but not included in STT
Scalability	Number of tenants is tunnel protocol dependent: VXLAN has a 24 bits long VNI ≈ 16000000, NVGRE also has a 24 bits long VSID and STT has a 64 bits long Context ID $\approx 1.8 \times 10^{19}$. DPS is the scalability limiting component.	Big number of EIDs and RLOCs possible. One RLOC address associated with multiple EIDs addresses. Issue with the MS and MR maximum information saved.	One PA associated with multiple CA. Suppress most of the control plane broadcasts messages and convert some of them in multicast messages.	Context ID fields is 64 bits long. Issue with the virtual switch which can not manage this much IDs.
Multi data center	Not specified but possible. It will require an important DPS to manage the whole network.	Yes even with non LISP data center	Yes as a site-to-site VPN. Each site must have a NVGRE gateway	Theoretically, yes as a site-to-site VPN. Practically, no because of the middle boxes issue

Table 5: Comparison of the Core isolation protocols

Core isolation	**PortLand**	**SEC2**	**VSITE**	**VNT**
Migration	Live migration thanks to gratuitous ARP. Possibility of lessening the number of lost packets with redirection.	Live migration thanks to gratuitous ARP.	Live migration if the VM stays in the same location otherwise offline migration.	Live migration. Based on TRILL or MLTP which uses RBridge nicknames for forwarding the messages so VM's IP or MAC address unchanged.
Resilience	Fat-tree topology induced resilience. Fabric manager back up even with slightly non identical information.	Multiple FEs. Backups of Central Controller (CC). Can uses a Distributed controller instead of CC.	Master/slave switches configuration with the virtual router redundancy protocol.	ECMP for multipath. Redundant multicast distribution tree. No centralized controller.
Scalability	Centralized controller. Huge stress on Fabric manager. Not scalable by default: $\approx$ 27000 hosts with 25 ARP request/second = Fabric Manager with 15 CPUs.	Centralized controller. Huge stress on Centralized Controller. Possibility to transform the CC in a Distributed Controller. Only 4096 VLANs by edge domain. Number of edge domain is not limited but depend on MAC address usage.	VLAN for isolation. Aggregation of multiple VMs under a locIP.	Multiple VMs MAC addresses aggregated under one RBridge nickname. VNI TAG (24 bits) allows for 16777216 virtual networks.
Multi data center	Not specified. Layer 2 Fat-tree topology to interconnect the core switches of each data centers and get one network spanning over multiple DC. Not really feasible in reality seeing the cost induced by the interconnection topology.	Multi domains by design but scalability issue, only 4096 VLANs per domain.	Multi sites by design but scalability issue, only 4096 VLANs per sites.	Ready for multi data center. When using TRILL it creates one big network with one control plane spanning over all the data centers. Whereas MLTP keeps each data center independent. Needs Layer 2 tunnels between data centers.

Core isolation	**802.1ad**	**802.1ah**	**VXLAN**
Migration	Need to allocate resources for the VLAN in the destination network ahead of time to have session continuity. Migration restricted to the same Layer 2 network.	Need to allocate resources for the VLAN in the destination network ahead of time to have session continuity. Migration restricted to the same Layer 2 network.	Need to allocate resources for the VXLAN in the destination network ahead of time to have session continuity. Migration across Layer 3 network possible.
Resilience	Link aggregation and switches redundancy.	Link aggregation and switches redundancy	VTEP in hypervisor so redundancy of server in order to migrate the VMs to a new server if the hypervisor is down.
Scalability	VLAN limit up to 16777216	VLAN limit up to $\approx 7 \times 10^{16}$	16777216 VNI possible.
Multi data center	Yes with the same VLANs on all data center.	Yes with the same VLANs on all data center.	Possible to use VXLAN as a site-to-site VPN with VTEP gateways.