A Case for Overlays in DCN Virtualization

Katherine Barabash[†], Rami Cohen[†], David Hadas[†], Vinit Jain[*], Renato Recio[*] and Benny Rochwerger[†]
[*]IBM
[†]IBM Haifa Research Lab

Abstract—**Server virtualization has brought about tremendous value to a modern computing landscape and in particular to data center and cloud infrastructures. Virtual server deployments have become ubiquitous in many development and production sites, in cloud infrastructures and in disaster recovery solutions. Network connectivity is a vital aspect of modern computing. In this work we explore the new requirements the server virtualization brings to the networking world and show how these requirements are different from those of a physical server connectivity. We then describe the Distributed Overlay Virtual Ethernet (DOVE) network architecture for building virtual networks infrastructure. We show how host-based overlays answer the novel networking requirements and describe a working example of their usage for network virtualization. We discuss the benefits and the drawbacks of the method, outline options for its efficient implementation, and show what additional work is required in order to base DCN virtualization on overlays.**

I. Introduction

Server virtualization is not a new concept. It has been around since mid 1960s when IBM Mainframe implemented the first virtual machine monitor for the 360/40 [1]. For several decades mainframe was the only virtualized platform, running vital enterprise workloads, achieving high rates of hardware consolidation, and providing high application availability and security.

Since the introduction, by VMware, of virtualization technologies for non strictly virtualizable commodity hardware architectures in the late 1990s, server virtualization ceased to be a privilege of high end enterprise clients and became possible on affordable off-the-shelf server hardware. This brought about abundant new virtualization use cases and scenarios and, eventually, led to a new paradigm of hosting and providing computing services, namely, cloud computing.

In recent years, multiple vendors are putting their efforts together in order to achieve efficient, secure, and convenient data center virtualization techniques. These efforts extend to providing hardware support, developing software infrastructures, creating unified management solutions. As a result, off-the-shelf virtualization technology today supports running enterprise level applications, hosting multiple independent tenants on shared infrastructure, and implementing advanced business continuity and load balancing scenarios.

Networking is an important enabling part of almost any computing infrastructure today. All the applications running in a data center need network connectivity to achieve the application goals, to access remote storage and services, to provide services to remote clients, etc. Traditionally, data center applications are deployed on physical servers and are interconnected according to the application needs, security, load balancing, and quality of service considerations. These and other considerations are very complex, making it hard and labor intensive to tailor custom configurations for every data center; moreover, maintaining and managing data center networks over time is a challenging task. To cope with the complexity, over the years, researchers and industry experts defined best practices for data center networking configuration and management [2], [3], [4]. In addition, protocols and tools for automatic configuration propagation, network equipment redundancy and failover were developed to help managing the network's dynamic requirements.

Initial use cases for the commodity virtualization platforms were desktop virtualization and creating development and testing environments for various uses [5]. Networking requirements of such scenarios were similar to the regular networking requirements. Naturally, network connectivity of virtual machines was approached as an extension of the existing and well understood networking paradigm. Virtual machine monitors and/or hypervisors were patched to provide network connectivity to virtual machines they host. Virtual machines became new end-points for the existing networking infrastructures; hypervisors became the edge switches for these new types of network end-points.

Today, when server virtualization technology has advanced to allow production data center deployments, many new application scenarios became possible. Examples of such scenarios are: elastic applications where application components are added and removed based on the application load; mobile application components relocated to different hosts based on hardware availability or distance to entities the component communicates to; site evacuation when all the application components are moved to another site before anticipated site service shutdown or disaster [6], [7].

In this work, we draw attention to the unique properties and requirements for interconnecting virtual servers in a data center. We stress the importance of satisfying these new requirements and explain why, in our opinion, host-based overlay networks is the future of virtual networking for data centers and cloud infrastructures. We describe one possible implementation of the host-based virtual networks and discuss benefits, drawbacks, and performance aspects of this solution.

A. Organization

The rest of this paper is organized as follows. In Section II we give our view on interconnecting virtual network endpoints and discuss why and how it is different from interconnecting regular physical servers. In Section III we describe the concept

978-0-9836283-2-3 © 2011 ITC

This paper was peer reviewed by subject matter experts for publication in the Proceedings of DC–CaVES 2011

of overlay, give examples of its current usages, and explain why overlay networks can be useful in a virtualized data center. In Section IV we present the Distributed Overlay Virtual Ethernet (DOVE) network architecture for interconnecting virtual machines (VMs). First, we describe this generic host-based overlay network architecture and then give an example of how this architecture was implemented to serve as federated cloud networking infrastructure of the FP7 EU project, RESERVOIR [8]. Conclusions and future directions are presented in Section V.

II. Networking in the Server Virtualization Era

On the surface, interconnecting virtual servers seems no different from interconnecting physical servers. Indeed, hypervisors are built in a way allowing virtual servers to run unmodified versions of operating systems and, hence, unmodified application stacks. This means that virtual servers can be viewed as regular network endpoints, not different from regular hardware servers. Virtual servers are configured with their own network addresses, e.g. MAC and IP addresses, and other networking attributes, e.g. IP subnet mask and default gateway IP address. Virtual servers participate in neighbor discovery protocols, e.g. ARP, and perform network endpoint routing just like regular servers do. Indeed, networking services for virtual servers can be provided by connecting them to the same physical network the hosting physical machine is connected to.

So why do we claim that there are important differences between interconnecting virtual and interconnecting physical servers? To answer this question, let us analyze the networking implications of consolidating DCN applications on virtualized platforms.

First, multi-component nature of modern data center applications led to a great increase in the amount of severs. Before virtualization, typical enterprise data center contained thousands of physical servers and this number was steadily rising leading to increasingly high costs in terms of floor space, power needs and management. With server virtualization, physical hosts are capable of hosting tens of virtual servers, thus making possible to satisfy application needs in server components without increasing the amount of physical servers in a data center. However, with the current tendency of treating virtual servers as physical network endpoints, the number of network endpoints continues to grow and can easily reach tens and hundreds of thousands per data center [9], [10], [11]. Network equipment and network management costs are rising with the increase in the number of interconnected endpoints, making data center networks larger, more complex, and harder to manage. DCN designs have to be reinforced with more switches or with switches with larger memories and learning capacities to cope with the amount of additional virtual MAC addresses that have to be learned.

Second, data center applications are created of clusters and/or communication tiers so that providing connectivity to a single application requires building several isolated communication links with varying quality of service, security, and management requirements. Due to application consolidation on virtualized platforms, data center networks must support significantly more different isolated networks than traditional data center networks do.

Third, as data center application loads depend on various external factors, resource requirements of applications and their components vary over time. In order to satisfy varying resource requirements without massively overprovisioning the physical resources, components of multiple applications are consolidated on shared physical hosts. In some cases, different application components running on a same physical host belong to different data center tenants. Thus, to add on a previous requirement, virtualized data center networks must support large amounts of isolated networks belonging to different tenants and, possibly, managed by different authorities.

Fourth, recent advances in the virtualization technology enable automating virtual machine lifecycle by allowing to create and to destroy virtual machines on demand, as well as to migrate virtual machines from one physical host to another. These capabilities allow for advanced use cases such as elastic applications and application component mobility. To take advantage of these capabilities and to enable these advanced use cases, virtualized data center networks must be very dynamic and support frequent addition, removal and moving around of endpoints. One important particular case is supporting virtual machine migration anywhere in a data center without having to reconfigure the virtual machine or cause it to loose its existing network connections.

Fifth, to add on to a previous requirement, virtualized data center must support deploying virtual machine anywhere in a data center, irrespectively of the underlying layout and configuration of the physical network components, so it can communicate as required by the application it is a part of. To achieve this, connectivity services for virtual machines must be independent and isolated from the underlying physical network. This means that changes in a virtual network does not have to cause physical network reconfigurations and vice versa. One important consequence of this requirement is that physical network configuration in a data center can remain static and continue to follow the time tested and convenient best practices.

A. Networking Requirements for Virtualized Data Centers

Here we summarize the discussion above and list the important new requirements server virtualization brings to the data center networking world. In addition to the regular requirements of interconnecting physical servers, network designs for virtualized data centers have to support:

- Huge number of endpoints. Today physical hosts can effectively run tens of virtual machines. With the increase of number or cores and IO capacity, a single physical machine will be able to host even more virtual machines.
- Large number of isolated and independent networks. The most important is to achieve address space isolation, management isolation, and configuration independence. Performance isolation is another important requirement.

- Multitenancy, where application components belonging to different tenants are collocated on a single physical host.
- Network and network endpoint dynamics. Server virtualization technology allows for dynamic and automatic creation, deletion and migration of virtual machines. Networks must support this function in a transparent fashion.
- Separation from the underlying physical network and isolation from changes in it.

These requirements call for a solution different from just extending physical networks up the hypervisors and connecting virtual servers to the physical data center network. What is required is proper network virtualization, i.e. creating virtual networks in a way similar to creating virtual servers: independent of physical infrastructure characteristics, isolated from each other, dynamic, configurable and manageable. This work describes our approach to this challenge, namely, using hypevisor based overlay to provide networking services to virtual servers in a data center.

III. Overlay Networks

Overlay networks is a method for building one network on top of another. Applications of overlay networks are many and the technology is well understood. Overlay networks are extensively used by research communities for Grid, HPC and distributed systems to create custom application level networks over infrastructures of the standard distributed components interconnected by standard network protocols. Industrial usage of overlay technology is abundant as well, e.g virtual private networks to achieve remote office and traveling employees connectivity [12], [13], [14].

The major advantage of overlay networks is their separation from the underlying infrastructure in terms of address spaces, protocols and management. In standard, TCP/IP networks, overlays are usually implemented by tunneling. Overlay network payload is encapsulated into headers and delivered over the underlying infrastructure. Overlay payload can be Ethernet, IP, other standard or custom application protocols. Underlay networks can use different technologies and protocols, can be homogeneous or heterogeneous, can be local to a site or an organization or public and wide, spanning the entire Internet. Encapsulation can be simple, aiming only to correctly deliver the overlay payload on top of the underlay infrastructure, and it can be complex, bearing lots of control information and allowing overlay nodes to relay and route packets among themselves.

The main drawback of the overlay techniques is the overhead it adds onto the packet processing in the end nodes and sometimes in the middle boxes. Apart from the direct costs of adding and parsing the encapsulating headers by the overlay nodes, there are additional non-direct performance penalties. Sometimes, the presence of encapsulation header causes packets to be dropped or to be handled by slow path processing units in switches, routers, and network appliances. In addition, encapsulated packets are larger in size than the original ones, potentially causing inefficiency due to unexpected fragmentation. Another drawback of using encapsulation in the data center is that encapsulated traffic becomes opaque to network monitoring tools and security appliances.

Abundance of overlay technologies and their deployments by providers, enterprises and research communities testifies that the need for overlay designs and their benefits outweigh the drawbacks. Network vendors support and introduce overlays in their products for widespread and proven use cases such as VPN, VPLS, OTV, etc. End point network stack designers create processing hooks to enable efficient creating, parsing and modifying of overlay headers on a fast packet processing path.

Recently, overlay network designs started to be used for interconnecting virtual machines. Most of the work in this space belongs to research community in distributed systems in attempts to utilize cloud platforms opportunities in building large scale distributed and peer to peer systems [15], [16]. In this work, we investigate the opportunities of employing the overlay networks technology to interconnect virtual machines in data centers. We show that this approach leads to satisfying the new networking requirements for virtualized data centers we have presented in Section II-A.

IV. Host-based Virtual Overlay Networks

The network virtualization architecture we propose employs the encapsulation technique from the overlay networks in order to achieve the separation of the virtual networks from the underlying infrastructure and from each other. The separation means separate address spaces, ensuring that virtual network traffic is seen only by network endpoints connected to this virtual network, and allowing different virtual networks to be managed by different administrators. In our architecture, overlay network nodes are located in physical hosts and are responsible for capturing virtual machines traffic and sending it between each other on the physical network. Overlay nodes do not relay encapsulated traffic between themselves as in peer to peer architectures. All the routing and forwarding in the physical network is relied upon the physical network infrastructure and, as such, builds upon the connectivity, security, and other properties the underlying network infrastructure provides.

A. Distributed Overlay Virtual Ethernet (DOVE) Architecture

We define the notion of Distributed Overlay Virtual Ethernet (DOVE) Network and allow administrators to manage DOVE Network instances in a data center. For example, DOVE instances can be created and deleted; virtual machines can be attached to and detached from DOVE instances. Upon creation, every DOVE instance is assigned unique identity and all the traffic sent over this overlay network will bear the DOVE instance identity in the encapsulation header in order to be delivered to the correct destination virtual machine. To achieve this, virtual machine interfaces are marked as being connected to a specific DOVE instance by the virtual networking component that resides in each physical host in a data center. We call this component Distributed Overlay Virtual

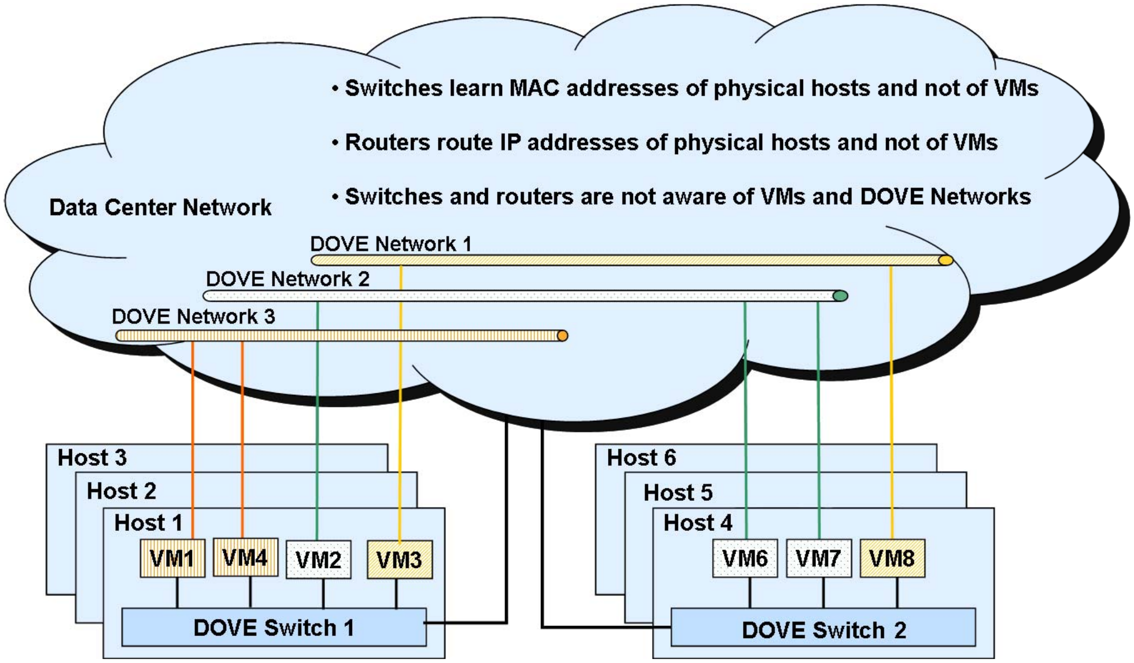

• Switches learn MAC addresses of physical hosts and not of VMs
• Routers route IP addresses of physical hosts and not of VMs
• Switches and routers are not aware of VMs and DOVE Networks
Data Center Network
DOVE Network 1
DOVE Network 2
DOVE Network 3
Host 3
Host 2
Host 1
VM1
VM4
VM2
VM3
DOVE Switch 1
Host 6
Host 5
Host 4
VM6
VM7
VM8
DOVE Switch 2

Fig. 1. Distributed Overlay Virtual Ethernet (DOVE) switches in physical hosts are responsible for creating connectivity between virtual machines: capturing their traffic, encapsulating it, and sending the encapsulated traffic over the physical data center network. Physical networking devices in a data center are not aware of virtual machines, their addresses and their connectivity patterns.

Ethernet (DOVE) switch (DOVE switch). DOVE switches are similar in function to the traditional hypervisor switches but have additional functions as overlay nodes.

First of all, DOVE switches mark network interfaces of the hosted virtual machines by assigning the virtual network identity to them. DOVE switches must sit in a network IO path of each hosted virtual machine and intercept traffic sent by it.

Second, upon getting the data packet originating in the hosted virtual machine, DOVE switch identifies the DOVE Network instance the packet belongs to. It then resolves the identity and the location of the target virtual machine and, if the destination is not hosted by the same physical host, encapsulates the packet with the header bearing encapsulation specific control information and sends it to the destination physical host over the underlying physical network.

Third, upon receiving the encapsulated packet from the physical network, DOVE switch parses and removes encapsulation header and delivers the packet to the correct destination virtual machine as identified both by the target virtual machine address in the packet and by the virtual network identifier in the encapsulation header.

In addition, DOVE switches participate in control plane protocols to exchange and distribute information about virtual machine location, virtual machine addresses, virtual machine migration events, etc.

Figure 1 shows DOVE switches residing in data center hosts and providing network service for hosted virtual machines so that virtual machines are connected to independent isolated overlay networks. As virtual machine traffic never leaves physical hosts in a non-encapsulated form, physical network devices are not aware of virtual machines, their addresses, and their connectivity patterns.

Different implementations of the described architectural concept are possible, depending on several factors:

- The kind of service provided to virtual machines. It is possible to provide a layer-2, e.g. Ethernet connectivity, layer-3, e.g. IPv4 connectivity, some custom application level protocols connectivity, etc. Providing Ethernet connectivity is beneficial in that it is simple and does not limit the upper layers of network stack in virtual machines. Providing IP connectivity to virtual machines is beneficial in that it allows to limit the amount of traffic sent to the physical network due to IP service protocols like DHCP, ARP, etc. On the other hand, choosing this type of overlay limits the ability to support applications requiring non-IP communications between virtual machines.
- Type of underlying physical network that is used as a carrier. Each implementation of host-based overlay architecture must assume a specific underlying physical network topology and technology and be tailored to it. More advanced implementations can be made capable of supporting different underlays, e.g. IPv4, MPLS, Internet, or even heterogeneous underlays.
- Whether the control plane is fully distributed or is relying on centralized components. Overlay nodes can be fully autonomous in discovering and distributing the control information or can rely on a centralized entity to keep and distribute it. Implementations can differ in the amount of control data sent between the overlay nodes, between the

overlay nodes and the controlling entities.

- Where in the host DOVE switches are implemented. The most straightforward option is to implement DOVE switch as a software component integrated into the physical server's network stack. Implementations involving forwarding hardware are possible as well.

While particular implementations can be different, all the solutions following the presented architectural concept will have the following desirable properties that are essential in order to support the advanced data center virtualization scenarios:

- Separation of virtual networks from the underlying infrastructure and from each other. Different virtual networks can be created, modified, migrated and deleted with no relation to one another and with no need to reconfigure the physical network components. Different virtual networks can be assigned address spaces independently of each other and of the physical addresses. Virtual machines are able to see network traffic of virtual networks they are connected to and are not able to see traffic of other virtual networks.
- Possibility of unrestricted placement and migration of virtual machines. Virtual machine placement and migration possibilities are not restricted by the physical network topology, so that virtual machines will continue to communicate when one or both of the communicating endpoints are migrated over the physical network switch, router, or site boundary.
- Configuration flexibility and manageability. Virtual network configuration is easy to manage and does not require involvement of physical network administrator. It is possible to integrate the virtual network management into the data center management tools and allow for transparent management of changes as required by modern elastic applications.

Of course, solutions based on the proposed architecture will inherit the drawbacks of the overlay network designs. Inefficiencies caused by these drawbacks must be resolved in order for the host-based overlay to scale into an acceptable industry strength solution, supporting huge number of network endpoints and large number of independent virtual networks.

- Per packet processing overhead can be reduced by smart tailoring of DOVE switch function into the physical server's networking stack, handling fragmentation, and by creating offload solutions. Encapsulation header standardization can reduce overheads incurred in middle boxes and the need for encapsulated packets to go through the slow processing path.
- Opaqueness of the inter-VM traffic to the traditional network monitoring tools and appliances can be overridden either by developing a new generation of these tools and appliances or by custom agents that will remove and recreate encapsulation headers at the key points of the packet path.

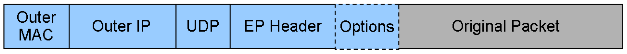


Fig. 2. DOVE packet encapsulation format.

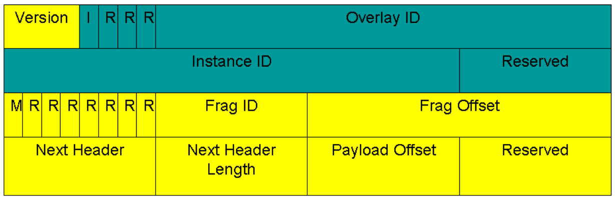


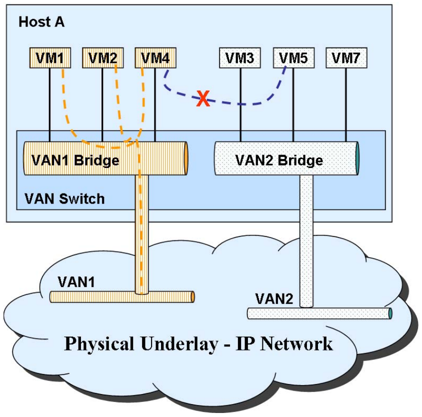
Fig. 3. DOVE encapsulation header. Proposed OTV extension fields are shown in yellow.

B. DOVE Encapsulation Header

Here we propose packet encapsulation format for DOVE that is based on and extends the Overlay Transport Virtualization (OTV) encapsulation format [12]. Basing the encapsulation format on an existing encapsulation format such as OTV allows for a common format for overlay network encapsulation. This encourages the switch ASIC and NIC vendors to add support which is essential for high performance implementations where encapsulation support is available through NIC offloads and in Gateways at the edge of Overlay Networks.

Figure 2 shows DOVE encapsulated packet, where the encapsulation header consists of an outer IP header followed by a UDP header that guards the DOVE encapsulation header. By using the outer IP and the UDP headers we get the benefit of ubiquitous acceptance of encapsulated packets throughout the networking infrastructure. Network forwarding and security devices will see these packets as standard and process them on their fast processing paths. Figure 3 zooms into the DOVE encapsulation header. As discussed previously, the Instance ID can be used to logically separate the overlay traffic of different DOVE instances in a common underlay. Our proposed extensions to the OTV header add support for versioning, fragmentation and addition of optional header extensions. As overlay networks evolve and get standardized, having versioning support will be critical to deploying the protocol header changes. While fragmentation should be discouraged wherever possible through adjustment of MTU in the physical infrastructure, we believe that fragmentation support will be required. Handing fragmentation at the IP layer is not preferred as administrators frequently configure middle-boxes to drop fragmented IP packets due to security reasons. Therefore we propose adding fragmentation support at the DOVE session layer where it will be transparent to network middle-boxes. Finally, the addition of optional header extensions will allow for accomodation of new requirements as overlay networks and use cases based on them continue to evolve. Figure 3 describes one possible extension to the OTV header that would enable fragmentation, versioning and the addition of future headers.

While we propose extending the OTV packet format for



VM Location Tables

VAN1

VM	MAC	Location
VM1	MAC1	local
VM2	MAC2	local
VM3	MAC2	Host B
VM4	MAC4	local

VAN2

VM	MAC	Location
VM5	MAC5	local
VM6	MAC6	Host C
VM7	MAC7	local
VM8	MAC8	Host B

Fig. 4. Edge bridges established for each VAN instance inside the physical host. VMs on same edge bridge communicate directly, while traffic out of the host is sent through overlay.

DOVE, the IS-IS control protocol used by OTV for address dessemination may not be suitable for DOVE as it is unlikely to scale to the limits where every Host is expected to have a DOVE switch. Considering the massive scaling requirements of DOVE networks, approaches where multiple centralized controllers are deployed to help in the collection and dessemination of address information seem preferrable. We see innovative evolution in this address dissemination approach.

C. Example of DOVE Architecture Implementation - RESERVOIR VANs

This section describes an example of a host-based overlay network implementation: virtual network infrastructure created as part of the EU project RESERVOIR [8] and referred to as Virtual Application Network (VAN). In RESERVOIR, VANs provide an Ethernet service to virtual machines by constructing an overlay between hosts and using a standard IP network as the underlying physical infrastructure. The solution is fully distributed, so that all the data plane and control plane functionality resides in host based modules. To virtual machines connected to a VAN instance, the VAN service is transparent and appears to be as though they are connected to peer virtual machines through an Ethernet network. All virtual machines belonging to the same VAN instance therefore belong to the same virtual layer-2, while virtual machines belonging to different VAN instances are isolated from each other.

Figure 4 presents a physical host, a VAN switch residing in it, and its hosted virtual machines. Hosted virtual machines are connected to two different VAN instances, VAN1 and VAN2. VAN switch implements two VAN instances in the host instantiated by two layer-2 bridges. All virtual machines belonging to the VAN instance are connected to an instance of bridge established for that VAN instance so that communication between two local VMs belonging to the same VAN instance is handled locally by the bridge without the packets having to enter the overlay.

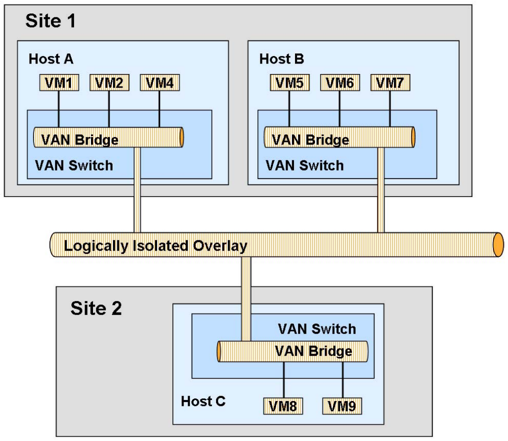


Fig. 5. VAN overlays provide connectivity between VMs on different hosts irrespective of their physical location. Traffic within the VAN instance is logically isolated through use of VAN instance identifiers in the VAN encapsulated headers.

The VAN switch maintains a table per VAN instance of destination VMs that local VMs are communicating with. When a VAN switch recognizes a packet destined for a VM that is not on the local host, it encapsulates the packet and includes the VAN instance id in the encapsulation header. The encapsulated packet is then addressed to the physical host where the destination VM is hosted and the physical IP underlay is used to send the packet.

VAN switches are aware of all virtual machines that are active on a particular VAN instance on that host. The VAN switch on the destination host recognizes the encapsulated packet and parses it to retrieve the VAN instance id. VAN switch then verifies that destination virtual machine is hosted on this host and belongs to the correct VAN instance. If everything is correct, the VAN switch removes the VAN encapsulation header and delivers the packet to the destination virtual machine. The use of a VAN instance id in the encapsulation header ensures logical separation between the VAN instances in the overlay. As shown in Figure 5, an overlay is constructed to provide VAN connectivity between VMs of the same VAN instance but hosted on different, possibly remotely located, hosts.

This mechanism can be used irrespectively of the physical location of the destination VM as long as physical connectivity exists between the hosts hosting the source and the destination. There may be situations where the physical underlay does not provide direct connectivity to hosts between different sites. This scenario is handled through the establishment of VAN proxies at each site with each VAN proxy having the ability to handle multiple VAN instances. The VAN proxies maintain a table per VAN instance of the destination VMs with the corresponding proxy that the VMs from the local site are communicating with. The VAN switch on the host encapsulates the packet and sends it to the local VAN proxy. The VAN proxy then delivers the packet to the destination site proxy based on its tables. The VAN proxy at the destination site then reconstructs the packet for delivery to the host that is hosting the destination VM. The use of VAN proxies is optional and is needed only when directly addressable physical connectivity does not exist between two sites.

Isolation between different VAN instances is maintained through the use of VAN instance id which provides a logical separation in the overlay. The VAN instance id is an essential part of the VAN encapsulation header and helps ensure the VAN switches provide connectivity to VMs in the same VAN instance while denying connectivity between VMs between different VAN instances.

A multicast or broadcast domain is created between the VAN switches residing on different hosts. When multicast support is not available, a multi-unicast may be used to reach all VAN switches serving a VAN instance. The VAN switches use this broadcast domain to flood a request to learn about unknown traffic for a VAN instance endpoint. Since VAN switches on hosts have complete awareness of VMs it serves for a VAN instance, the VAN switch hosting the endpoint responds to the learn request.

When a VM migrates or is shutdown, the VAN switch where it was previously hosted updates its tables to reflect that the VM is no longer active on this host. When a VAN switch on the host where the VM was previously operational sees traffic destined for this migrated VM, it will send an un-learn request to the VAN switch originating this traffic. This enables the VAN switch on the originating host to re-initiate a learn request. One possible optimization is for a VAN switch to send immediate un-learn requests to all the VAN switches the previously active VM was communicating with.

In RESERVOIR, VAN switches were implemented in KVM hypervisor and the resulting system was deployed and run in a setup consisting of two remote data centers with several x86 boxes each. For test purposes, RESERVOIR team has deployed multi-tier web serving applications and exercised elasticity and migration scenarios. More details on KVM implementation and its performance characteristics can be found in [17]; details on a more advanced scenario using VANs for deploying virtual networks over autonomous federated cloud infrastructures can be found in [18].

V. Conclusions

In this work we have analyzed the advanced novel data center scenarios that have became possible due to server virtualization and derived new networking requirements these scenarios create. To allow server virtualization to succeed in data center and cloud infrastructures, there is a need to support creating, configuring and maintaining large numbers of isolated, efficient, and dynamic networks connecting virtual machines. Moreover, there is a need to enable these networks to be managed independently of each other and of the underlying infrastructure and to have independent, possibly overlapping address spaces.

We have discussed the challenges and possible solution directions and presented their properties. We have described the Distributed Overlay Virtual Ethernet (DOVE) Network Architecture as one possible solution. This architecture allows creating virtual networks that are independent from the underlying physical infrastructures and each other, can be separately managed and configured, have independent address spaces and are highly dynamic. The proposed architecture consists of networking modules in physical hosts and a control infrastructure. Host based networking modules are responsible for intercepting the virtual machines packets, resolving the destination virtual machine location, and, if the destination is outside the source physical host, encapsulating the packets and sending them over the underlying physical network to the destination physical host. In the destination host, the networking module is responsible for parsing and removing the encapsulating header and delivering the packets to the destination virtual machine. The architecture supports mechanisms to handle connectivity between the hosted virtual machines, external network connectivity, virtual machine migration, address provisioning and address resolution.

In addition to the generic DOVE Network Architecture, we have described one particular implementation of network virtualization framework that goes along the lines of this architecture, RESERVOIR VANs. VANs solution provides isolated Ethernet networking service to virtual machines over the standard IP physical infrastructure using a fully distributed control plane. DOVE Architecture based solution providing layer-3 (IP) service to its clients is currently being developed by us. This new solution, layer-3 Distributed Overlay Virtual Ethernet (DOVE) Network, is a next step towards a proper network virtualization in a data center.

There are many challenges that need to be resolved in order to make overlay based solutions practical. First of all, it is not straightforward to provide efficient implementations of host-based networking components so that the processing per packet overhead will be minimal. In addition, non-standard encapsulation header can cause problems for traditional network appliances, network offload engines, and traffic monitoring tools. Thus, standardization is vital in order for the solution to become practical. Control plane developments are crucial as well: protocol development and standardization will help implementing coherent integrated data center management solutions; control plane scalability, coherency and performance are important challenges for future work.

In summary, although there is a lot of work ahead, we believe that the future of data center networking is in hypevisor-based overlay architectures. We continue to advance the DOVE Network Architecture and make solutions based on it a reality.

Acknowledgment

The authors would like to thank the RESERVOIR networking infrastructure team, in particular Irit Loy, Sergey Guenender, Liran Schour, and Kenneth Nagin for their work on developing the implementation of VAN overlay and advanced scenarios based on it. We also thank Alexander Landau and Edi Shmueli for their contribution to the performance evaluation and optimization of VAN implementation.

References

[1] J. P. Buzen and U. O. Gagliardi, "The evolution of virtual machine architecture," in *Proceedings of the June 4-8, 1973, national computer conference and exposition*, ser. AFIPS '73. New York, NY, USA: ACM, 1973, pp. 291–299. [Online]. Available: http://doi.acm.org/10.1145/1499586.1499667

[2] Cisco data center network architecture. [Online]. Available: http://www.cisco.com/go/datacenter

[3] Brocade data center best paractices. [Online]. Available: http://www.brocade.com/data-center-best-practices/index.page

[4] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," in *Proceedings of the ACM SIGCOMM 2008 conference on Data communication*, ser. SIGCOMM '08. New York, NY, USA: ACM, 2008, pp. 63–74. [Online]. Available: http://doi.acm.org/10.1145/1402958.1402967

[5] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the art of virtualization," in *Proceedings of the nineteenth ACM symposium on Operating systems principles*, ser. SOSP '03. New York, NY, USA: ACM, 2003, pp. 164–177. [Online]. Available: http://doi.acm.org/10.1145/945445.945462

[6] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the clouds: A berkeley view of cloud computing," Feb 2009. [Online]. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html

[7] (2011) How cloud computing will change application platforms. [Online]. Available: http://blogs.forrester.com/john_r_rymer/11-04-26-how_cloud_computing_will_change_application_platforms

[8] B. Rochwerger, D. Breitgand, E. Levy, A. Galis, K. Nagin, I. M. Llorente, R. Montero, Y. Wolfsthal, E. Elmroth, J. Caceres, M. Ben-Yehuda, W. Emmerich, and F. Gal, "The reservoir model and architecture for open federated cloud computing," pp. 1–11, 2009.

[9] R. Miller. (2009) Who has the most web servers? [Online]. Available: http://www.datacenterknowledge.com/archives/2009/05/14/whos-got-the-most-web-servers/

[10] C. Harris. (2011) Data centers face growth challenges. [Online]. Available: http://www.informationweek.com/news/hardware/data_centers/229600034

[11] R. Figueiredo, P. A. Dinda, and J. Fortes, "Guest editors' introduction: Resource virtualization renaissance," *Computer*, vol. 38, pp. 28–31, May 2005. [Online]. Available: http://portal.acm.org/citation.cfm?id=1069588.1069631

[12] H. Grover, D. Rao, D. Farinacci, and V. Moreno, "Overlay transport virtualization," draft-hasmit-otv-03, Jul. 2011.

[13] R. Callon and M. Suzuki, "A framework for layer 3 provider-provisioned virtual private networks (ppvpns)," RFC4110, Jul. 2005.

[14] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, "Resilient overlay networks," 2001.

[15] A. Ganguly, A. Agrawal, P. O. Boykin, and R. Figueiredo, "Wow: Self-organizing wide area overlay networks of virtual workstations," in *In Proc. of the 15th International Symposium on High-Performance Distributed Computing (HPDC-15*, 2006, pp. 30–41.

[16] ——, "Ip over p2p: Enabling self-configuring virtual ip networks for grid computing," in *In Proc. of 20th International Parallel and Distributed Processing Symposium (IPDPS-2006*, 2006, pp. 1–10.

[17] A. Landau, D. Hadas, and M. Ben-Yehuda, "Plugging the hypervisor abstraction leaks caused by virtual networking," in *SYSTOR 2010: The 3rd Annual Haifa Experimental Systems Conference*, 2010.

[18] D. Hadas, S. Guenender, and B. Rochwerger, "Virtual network services for federated cloud computing," IBM Research Division, HRL, Tech. Rep. H-0269, Nov. 2009.