Enabling Consistent Policies in Networks with Physical and Virtual Servers

Martin Diviš

mdivis@cisco.com

Cisco Systems (Czech Republic) s.r.o.
Prague, Czech Republic

Abstract

Server and desktop virtualization offer many advanced features not available in traditional hardware based computing environment. It also presents disruption to the network environment in the area of policy enforcement and administrative responsibilities. In many cases, this slows down users on the way to the massive virtual computing environments. There are emerging standards under ratification in IEEE whose aim is to resolve the described conflict. This article describes those standards as well as possible future developments in this area.

Keywords: virtualization, policy, management, data center, IEEE.

1 Virtual Computing Environment and Networking

Virtual computing environment is a generally adopted technology in today's data centre environment which enables better utilization of traditionally underutilized server infrastructure and enables new feature not available on bare hardware infrastructure, such as virtual server mobility from one physical server to another while the server is running, high availability through virtual machine state replication and instant failover and many others.

What makes this possible is implementation of virtual and abstracted software implementation of what is normally physical hardware within which the virtual machine is executed without noticing significant difference from physical hardware. This software component is referred to as **hypervisor**. Hypervisor's role is to create virtual hardware for each virtual machine (VM), but also ensure proper access of virtual machines to physical resources, such as CPU, RAM, block I/O devices, USB devices, multimedia devices, and network interfaces.

From the point of view of security, this presents a whole range of new potential security issues:

- How do we know that one VM will not gain access to resources of another VM? For example, read memory of another VM, get access to disk of another VM?
- How do we ensure that one VM will not overload physical hardware thus preventing other VMs to run properly?
- How are we going to apply security policies in a shared network environment?
- How do we make sure hypervisor cannot be compromised from within VM thus putting at risk all other VMs running on that hypervisor?

Some issues have to be solved within hypervisor directly, for some, hypervisor can use hardware resources (examples being memory access or privileged modes of execution provided by latest CPU architectures), but some are not that easily overcome as they cannot be easily solved by hypervisor alone. Classic example

is networking and application of networking policies, including security policies, within environment which is partially virtual and partially physical.

This article will focus on issues in networking environment and the ways how the issues can be solved.

1.1 Current Status

Networking in today's virtual computing environment is complex. The complexity comes from the fact that networking spans multiple physical components with separate management and in case of virtual computing environments, even with multiple teams managing the network.

Traditional datacenter network follows hierarchical structure of core, aggregation and access layers with specific functions:

- **Core layer** of the network connects potentially multiple aggregation blocks.
- **Aggregation layer** of the network provides L2/L3 boundary and typically provides also network services such as security devices – firewalls, intrusion detection/protection systems – and application-related services - server loadbalancing, application acceleration.
- **Access layer** of the network, where servers are connected to the network.

Network policies are then enforced partially in the access layer – assignment to VLANs, QoS marking, server specific access lists – partially in the aggregation layer.

1.2 Issues

Virtual computing environment brings a significant change to this situation, as the implementation of the network I/O virtualization is based on software implementation of 802.1Q compliant bridge within the hypervisor, which presents another layer of networking and therefore another layer of management.

There are a few issues resulting from this situation:

- Management of virtual environment is not under control of team of network specialists and even less security specialists, but rather under control of server specialists or virtualization specialists.
- Features of the network devices are greatly inconsistent and therefore application of equal network policies for physical and virtual computing environments is difficult and sometimes impossible.

What practical consequences may this situation have? Imagine two VMs communicating with each other within single hypervisor. Since hypervisor contains 802.1Q compliant bridge, it will bridge network traffic between those VMs directly. Any policy set in the physical network will not be taken into account since the traffic will not pass any of the policy enforcement points within the network. The situation is shown on the diagram 1.

Another example might be multiple machines communicating outside of the physical server – how to distinguish between the frame from VM A and VM B? Based on IP addresses? Based on MAC addresses? MAC address is inconvenient and may change if new VM is created for the same workload. IP address can be spoofed. We no longer have the physical port where we could lock a physical server with specific IP address or MAC address based on learning to that port and ensure its authenticity. This affects QoS settings, security policies, server to VLAN assignments etc.

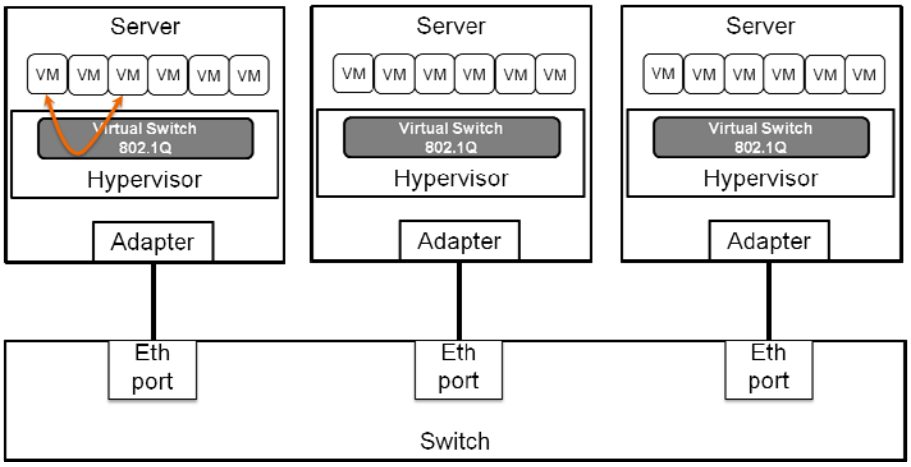


Diagram 1: Local switching of VM to VM traffic on a single hypervisor.

1.3 Characteristics of Desired Solution

Ideal solution to the problem would have following characteristics:

- Enable consistent policy model across physical and virtual environments
- Maintain the management model with traditional role separation between specialists for networking, security, and server infrastructure
- Keep the undisputable advantages of virtual computing environment, such as fast server deployments, and VM mobility and availability features

2 Approach to the solution

It appears that one successful approach to the problem might be extension of the network-managed port closer to the Virtual Network Interface Controller (vNIC) of the VM and in this way to mimic the structure of physical switches and physical servers – **one vNIC is connected to exactly one virtual Ethernet port** in the same way as one NIC of a physical server is connected to exactly one physical network port.

In this way, we have back the port which can be managed and to which policies might be applied.

However, from the practical perspective, it would be highly inconvenient for the virtual infrastructure manager to ask network manager for a new virtual server port each time new VM is created (as it is the case in case of physical server deployments). This would limit one of the principal advantages of computing virtualization.

Second component of the solution therefore is **split between policy definition and enforcement** – which is what interests networking manager and security manager primarily – **from the policy application process**. Imagine each of the virtual ports being deployed inheriting set of policies from a pre-defined profile based on a workload being planned for the newly deployed VM. This is a practical feature from another perspective as well – if we need to change certain policy, we can do it on profile level rather than having to change the policies on all ports individually.

Are there other issues to solve? Yes there are – we need to consider VM mobility. We have a virtual switch port to which vNIC is connected. If we move the VM with its vNIC to another physical server, the virtual switch port to which the vNIC gets connected on the target physical server must be configured identically to the one on the source physical server. Or better – we need to move the virtual switch port with the VM – after all, it's virtual, isn't it?

So the third component to the solution is **virtual switch port mobility**, which must be coordinated with the management infrastructure of the virtual computing environment.

2.1 Making the virtual port

There are obviously several ways to implement such networking environment.

One approach is software implementation of what may resemble modular switch with switch interface modules being replaced by hypervisor-embedded virtual interface modules and supervisors of a modular switch being replaced by (virtual) servers running the control plane functions of a supervisor.

Such implementation exist today such as embedded Distributed Virtual Switch (DVS) by VMware as part of vSphere 4.0 and higher or Nexus 1000V by Cisco for vSphere 4.0 providing additional features on top of standard DVS. Logical and physical view of DVS is shown on a diagram 2.

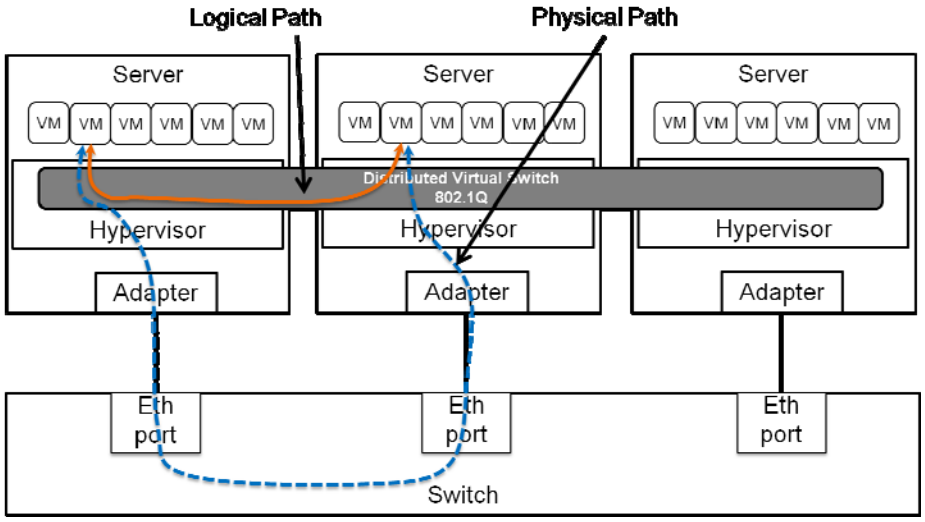


Diagram 2: Logical view of Distributed Virtual Switch.

Implementation of such distributed switch brings benefits especially in terms of management responsibilities separation, switch features, and support for virtual port mobility. Virtual port can be removed from one virtual interface module and placed to another one as the VM moves from one physical server to another. With port, policies follow the VM.

On the other hand, this still does not bring the whole environment into the same level of manageability and policy consistency as the purely physical environment. There is still another separate switching layer (virtual switch) and therefore separate management entity and perhaps different capability set within the different switching layers.

From this point of view, the proper solution would be to make all vNICs of VMs connected directly to the same physical switch to which the physical servers connect to. This way, network policies could be consistently applied on switch interfaces for both physical and virtual hosts. However, this is not easily done since there is usually high number of VMs running on a single hypervisor and small number of physical Ethernet interfaces in the server aggregating all inbound and outbound communication, therefore one to one pairing of vNIC to physical switch port is not economically efficient.

Potential benefits of such solution are, however, so attractive, that there are currently two IEEE standardization streams being backed up by major vendors in the area of networking and virtualization in

works. Those proposed standards present different levels of solving the networking issues for virtual computing environments and both stanrards - 802.1Qbg and 802.1Qbh - extend the relatively new family of Ethernet standards for so called Data Center Bridging (DCB).

2.2 802.1Qbg – Edge Virtual Bridging

Edge virtual bridging is based on simple idea - offloading of the bridging function embedded in the hypervisor as an virtual switch into the hardware switch to which the physical server is connected. The principle of the solution is shown at the diagram 3.

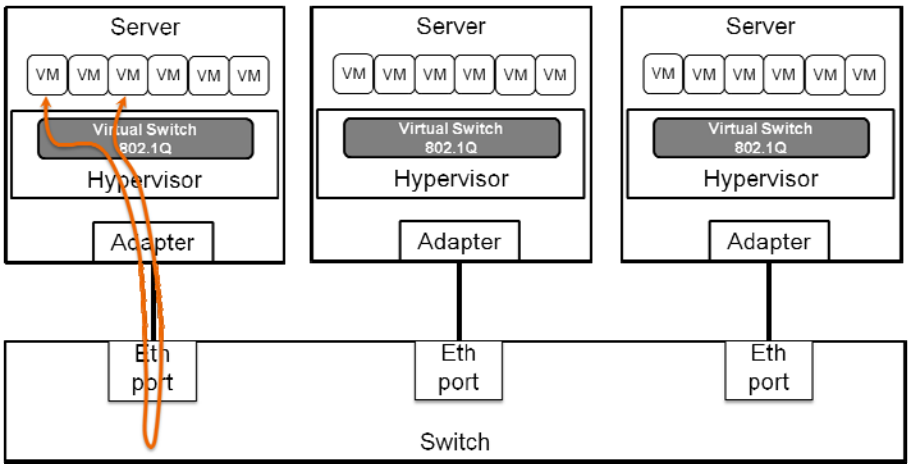


Diagram 3: Edge Virtual Switching – Reflective Relay.

All outbound Ethernet frames from the VM are sent via uplink to the first adjacent physical switch, where they are forwarded based on their destination MAC address. There is, however, one difference from the standard switch behavior – should the destination MAC address be on an hypervisor, from which the frame arrived at the physical switch port, it is forwarded on the same port back. This is different from the standard 802.1Q which prohibits frame forwarding to a source port of the frame and requires change in the switch hardware or firmware.

More important is the fact we gain very little in terms of policy application and enforcement. As discussed earlier, virtual port is needed as an entity for policy application and enforcement on one to one basis with vNICs of VMs. Reflective Relay can provide such function, but the virtual ports are created based on MAC addresses of the vNICs (frame source MAC). For each vNIC's MAC address physical server sees, it will create virtual port which can be managed.

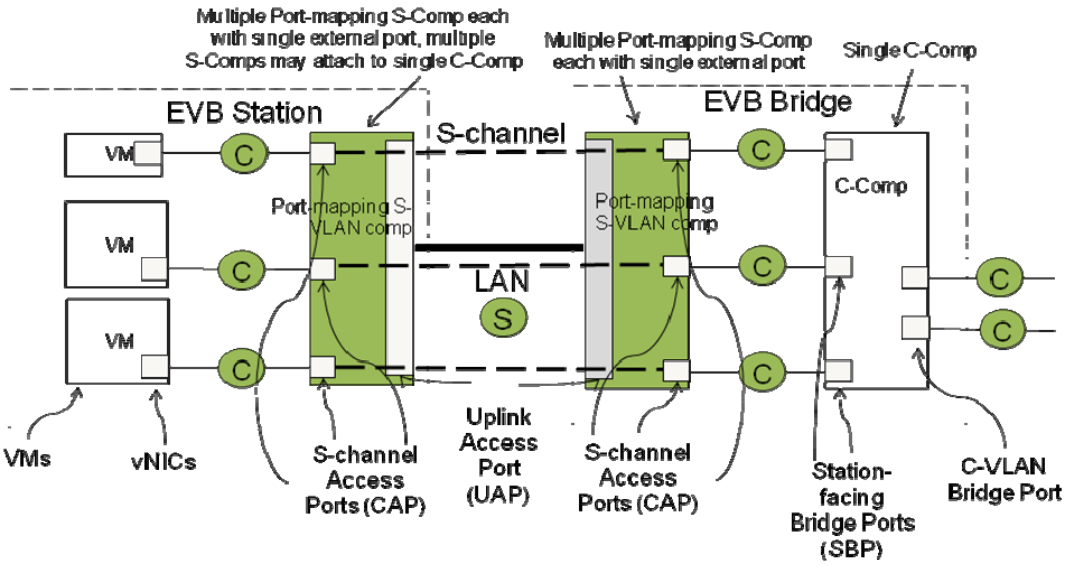
This is not perfect solutions – MAC addresses can be changed or spoofed or even not known – imagine implementation of transparent firewall within a VM, which does not change MAC address of the frames it sees – its port would be not identifiable. Therefore, the standard proposes for a new protocol – Virtual Station Interface Discovery Protocol (VDP), which is responsible for communication with the hypervisor for the guaranteed assignment of MAC to a particular VM. VDP is crucial protocol and its guaranteed delivery is ensured by another new protocol – Edge Control Protocol (ECP) defined within the standard for reliable transport. Thus, it requires modification of each hypervisor the Reflective Relay is connected to.

Standard 802.1Qbg therefore goes further and defines so called Multichannel behavior.

Multichannel behavior relies on so called S-channels. For each vNIC of each VM, S-channel is allocated, which effectively means that all outbound frames from vNIC are tagged by so called S-TAG, which

uniquely identifies the vNIC and therefore virtual switch port. On adjacent physical switch (Edge Virtual Bridge in the terminology of the standard), frame is mapped onto a virtual port which is then managed as any other physical switch port.

Principle of Multichannel behavior is shown on diagram 4, where EVB station is a physical server running hypervisor and VMs. Port-mapping component is responsible for adding S-TAG to an outbound frame and S-TAG stripping for inbound frames. Correspondingly, EVB bridge is the adjacent physical switch, also having Port-mapping component providing the same function.



Multichannel capability within proposed standard 802.1Qbg present significant advancement in consistent policy enablement, however, it's scalability limits are in practice serious.

Basis for this statement is mainly the way datacenter server infrastructure has evolved in the recent years with the rapid increase in deployment of blade server solutions. Blade server solution consists of blade chassis, which includes the chassis itself, power supplies, fans, chassis management and slots for Ethernet, Fibre Channel, and other switches and slots for server blades. In a way, it looks like a small autonomous rack with servers and LAN and SAN switching. The issue is, that the Ethernet switch is the EVB bridge for all blade servers within the chassis and therefore is able to switch frames only for the VMs running on blade servers within the chassis. This is clearly not scalable.

Another practical limitation is the fact that the switches embedded within the blade chassis do not often provide the same set of capabilities as upstream access of aggregation switches, therefore this solution does not have to be proper for consistent policy application and enforcement.

This is the target of second IEEE standard proposal for Virtual Bride Local Area Networks enhancement – standard 802.1Qbh.

2.3 802.1Qbh – Bridge Port Extension

Proposed standard 802.1Qbh brings wider view on the concept of virtual switch ports through so called Port Extension concept. Port Extender is a device (hardware or software), which attaches to a traditional physical switch port of a 802.1Q bridge, which then provides logical (or virtual) ports that are fully manageable ports of the 802.1Q bridge. This 802.1Q bridge is then called Controlling Bridge.

Moreover, Port Extender device may be cascaded, thus single Controlling Bridge may control tens of Port Extenders having tens of ports each, still behaving as a single 802.1Q bridge. Traffic from each port of

each Port Extender is reliable separated from other traffic using so called E-channels, which, like S-channels use S-TAG for frame tagging, use E-channel identifiers (ECID) for tagging of frames from and to extended ports.

For the switch, this means there is no reliance on MAC address related to virtual port identity and the switch may also learn the MAC addresses in the usual way.

The principle of tagging and forwarding of frames by controlling bridge and port extenders in cascaded topology is shown on diagram 5.

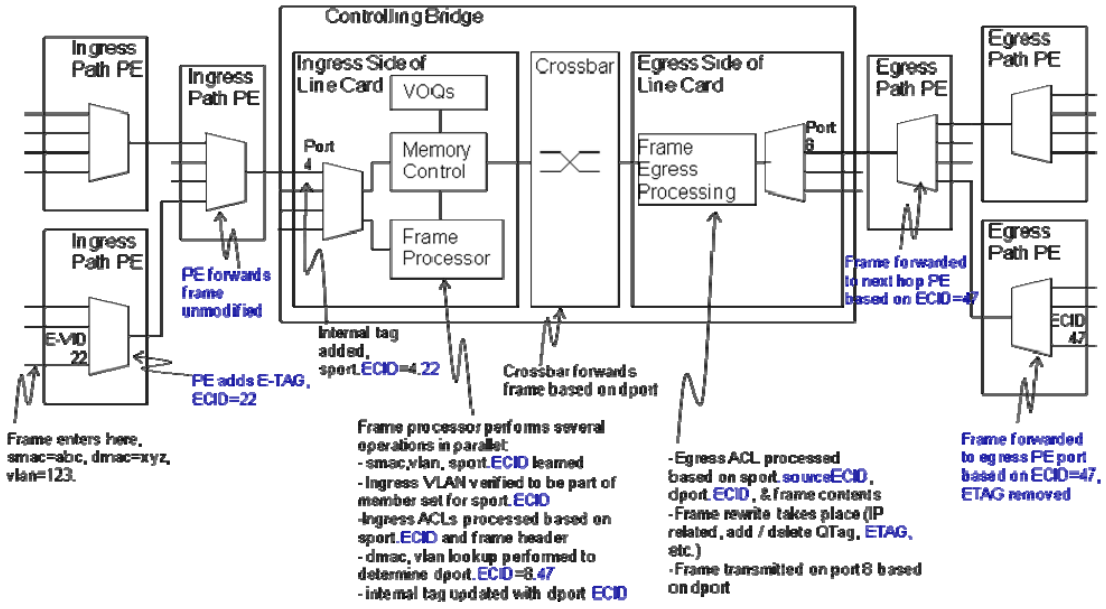


Diagram 5: Controlling Bridge and Port Extenders, frame tagging.

Port extenders forward traffic based on:

- Destination MAC address when frame travels from end-host port to the controlling bridge.
- Destination ECID when they travel from controlling bridge to a host port.

Port extender has for efficiency reasons two forwarding tables, one for unicast traffic, which contains one entry per destination ECID and points to an upstream port (port facing host port in the cascade), and one for multicast table, which contains multicast ECIDs for each multicast group with bitmasks of Extended and Cascaded ports to be used for multicast replication. In this way, controlling bridge does not have to perform multicast replication for all hosts which are members of multicast group, but the replication is distributed.

ECID have 14 bits long, where:

- ECID 1-4095 are reserved for Extended Port identificators,
- ECID 4097-16382 are reserved for multicast groups,
- Values 0 and 16383 are reserved for internal purposes.

This in effect means that single controlling bridge may control port extenders with total of 4094 ports and since the major goal is to support connectivity of VMs, one controlling bridge may control ports for about 4000 virtual machines.

Examples of unicast and multicast forwarding tables of port extenders are shown at diagram 6.

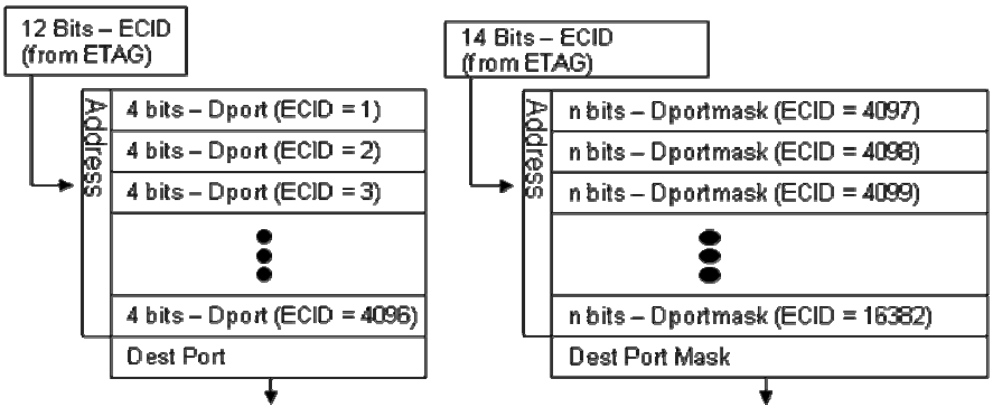


Diagram 6: Unicast and multicast forwarding tables of a port extender.

The full header of E-channel frame looks like the one on diagram 7. Ethertype is copied from Ethernet frame, PCP and DEI serve for traffic class selection, Ingress ECID is 12 bits long since it is always identification of an extended port, destination port is 12 bits for unicast frame and 14 bits for multicast frame.

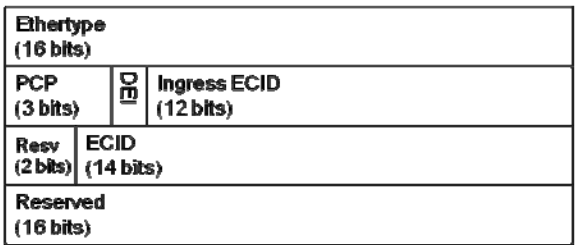


Diagram 7: E-channel header format.

As of today, there are several practical implementations of pre-standard 802.1Qbh controlling bridges and port extenders, at this stage all by Cisco Systems. Nexus 7000 and Nexus 5000/5500 switches implement roles of controlling bridges while port extender functions are implemented in the Nexus 2000 Fabric Extender range and in virtualization port adapters for server platforms. Implementation of the pre-standard by other vendors is in the works.

2.4 Making it work with the hypervisor

We have seen above that there are new standardization efforts underway targeted on making Ethernet bridging for virtual computing environment being equivalent in terms of management and policy application equivalent to the traditional physical one. However, as stated above, virtual computing environment brings significant benefits in its added value features line virtual server mobility, high availability, and others.

For this reason, good solution building on the standards mentioned above cannot exist alone, but needs to be integrated with hypervisor management. The minimum integration level is:

- Hypervisor management tools have to have access to bridging environment management tools to the pre-defined policies or profiles, which are then ready for use by the hypervisor manager.
- Hypervisor management tools have to have access to bridging environment management tools to signal mobility actions, so the bridging environment can synchronize extended port movement from one port extender to another with the movement of VM.

This means that though the extended ports bring benefits on their own and can be used in theory with any hypervisor, without integration with hypervisors or hypervisors management tools, the solution would be restricting virtual computing environment capabilities.

Therefore, it is important integration API's do exists on both sides making the integration possible and not restricting the solution for certain products only

3 Conclusion

As we have seen, virtual computing environment significantly changes the way we look at the traditional hierarchical design of data center LAN infrastructure and makes policy management and enforcement more difficult and inconsistent with physical world.

We have also seen several approaches to the problem from purely software ones to those based on hardware offload of Ethernet bridging.

While the hardware solution is clearly technically superior by offering the same capabilities of the solution in software and providing more consistency with the physical LAN management, only time will tell what solution will be accepted by the market and thus get widely adopted.

References

[1] *IEEE 802.1Q – Virtual LANs*. http://www.ieee802.org/1/pages/802.1Q.html

[2] *IEEE 802.1D – MAC Bridges*. http://www.ieee802.org/1/pages/802.1D-2003.html

[3] *IEEE 802.1Qbc – Provider Bridging: Remote Customer Service Interfaces* http://www.ieee802.org/1/pages/802.1bc.html

[4] *IEEE 802.1Qbh – Bridge Port Extension*. http://www.ieee802.org/1/pages/802.1bh.html

[5] *IEEE 802.1AE – MAC Security*. http://www.ieee802.org/1/pages/802.1ae.html

[6] *IEEE 802.1Qbg – Edge Virtual Bridging. http://www.ieee802.org/1/pages/802.1bg.html*