Hyper-V Architecture and Networking Considerations for Small to Medium Businesses

Dell Storage Engineering
May 2014

A Dell Technical White Paper

Revisions

Date	Description
May 2014	Initial release

Table of contents

Executive Summary

This paper will examine all of the various components which go into Hyper-V virtualization on a Dell EqualLogic storage platform. It will discuss Windows Server Hyper-V architecture with a particular focus on networking technologies. It will examine the various storage interface options (iSCSI, Fibre Channel, and SAS) available to a Hyper-V host, and will conclude by providing three example network design templates particularly tailored for a Small Medium Business.

Audience

This technical guide provides an overview for an Information Technology administrator looking to gain a better understanding of the technologies surrounding Windows Server Hyper-V

This paper assumes that the reader has:

- Previous Windows Server administration experience including installation and configuration
- Understanding of virtualization technologies
- Some familiarity with Microsoft PowerShell or some other scripting language

Software information

The following table shows the relevant software and firmware versions discussed in this paper.

Vendor	Model	Software Revision
Microsoft	Windows Server 2012	NA
Microsoft	Windows Server 2012 R2	NA

1 Introduction

Hyper-V is the primary engine that drives Windows Server 2012 "beyond virtualization" initiative. The primary responsibility of Windows Server 2012 Hyper-V is to provide the tool kit that organizations will use to create a shared pool of compute, network, and storage resources where servers and applications can be virtualized for consolidation, scalability, and mobility purposes.

This paper examines the key components of Hyper-V architecture and then the details surrounding Hyper-V networking. Administrators need to have a solid understanding of these concepts prior to implementing a Hyper-V solution into their environments.

2 Windows Server 2012 installation and licensing options for Hyper-V

There are two options when installing Windows Server 2012. The Windows Server full version has a GUI and the Server Core installation does not. . With Server Core, the administrator uses the command line and PowerShell to configure and administer the server. Server Core includes a basic set of services and features. Administrators can add features "on-demand" using PowerShell to support common infrastructure roles such as domain controllers, failover clustering, DNS, and Hyper-V. The primary advantage of using server core is that it consumes less CPU cycles and has a smaller disk space and memory footprint. For example, a typical full Windows Server 2012 OS will consume 8 -10 GB of memory. A Server Core installation will consume about 3 - 4 GB of memory. In addition, with fewer running services, studies have shown that Server Core will use up to 25% less CPU cycles. Server Core also increases security as it provides a smaller attack footprint since it has far fewer running processes. Server Core uses Datacenter and Standard licensing schemes. Unless there is a particular need for the GUI, Microsoft recommends Server Core as the default Windows Server 2012 installation option. The GUI can be added in as a feature using the PowerShell if needed.

Windows Server 2012 was released with a new licensing model which includes two primary options: Datacenter and Standard. Aside from prices, the primary difference between the two options is the amount of visualized instances. With a standard license, the host server can have up to two virtualized instances (VMs). With Data Center, a user gets unlimited virtualized instances.

Table 1 The following table summarizes Windows 2012 Licensing Options:

Edition	Features	Licensing model
Datacenter (GUI and Core)	Full Windows Server with unlimited virtual instances	Two processors/License + Client Access Licenses (CAL)
Standard (GUI and Core)	Full Windows Server with two virtual instances maximum	Two Processors/License + Client Access Licenses (CAL)
Hyper-V Server*	Server Core with Hyper-V only, with no virtual instances.	Free but licensing is applied to individual virtual instances

***** Microsoft Hyper-V Server is a dedicated, no-cost, standalone product that contains the hypervisor, Windows Server driver model, virtualization capabilities, and supporting components such as failover clustering It does not contain a GUI or other features and roles that the full Windows Server operating system has.

Note: Microsoft Hyper-V Server 2012 does not include any guest operating system licenses. Customers who wish to download and use Microsoft Hyper-V Server 2012 for their key workloads need to license their virtualized workloads accordingly with either Standard or Datacenter licenses.

For more information on Windows Server 2012 Installation options, go to http://technet.microsoft.com/en-us/library/hh831786.aspx

3 Hyper-V architecture

A common misconception about Hyper-V is that because it is necessary to install Windows Server 2012 prior to installing Hyper-V, it is a Type-2 virtualization where the hypervisor solution sits on top of the underlying operating system. However, like VMware ESX and Citrix XenServer, Hyper-V is a true Type 1 virtualization solution.

Beginning with Windows Server 2008, the Microsoft Hyper-V hypervisor uses a virtualization solution called Hardware Virtualization. Hardware Virtualization leverages virtualization features built into the latest generation of CPUs from both Intel and AMD. These technologies, known as Intel VT-x and AMD-V respectively, provide a hardware assisted virtualization layer (HAVL) below ring 0 known as ring -1. By using ring -1, hypervisors can operate by communicating directly with the system hardware resources, essentially leaving ring 0 available for host and guest operating systems.

Note: Windows Server 2012 Hyper-V requires that host processors support Hardware Virtualization technologies. Since 2006, most Intel processors include VT-x and AMD processors include AMD-V technologies. All Dell 11g and 12g servers use processors that are Hardware Virtualization ready.

The hypervisor is the core component of Hyper-V. In Windows Server 2012, the hypervisor is created when the Hyper-V role is installed and enabled. Installation of the Hyper-V role requires the host to reboot twice. During this process, the Hyper-V hypervisor is created and slipped underneath the Windows Server 2012 installation to run on ring -1. This location sits between the physical server hardware layer and the host operating system and any Hyper-V virtual machines (VMs). The Hyper-V hypervisor then presents independent, isolated virtualized hardware environments for the host operating system and created virtual machines called partitions. After the Hyper-V role is installed on the host (also now referred to as nodes or virtualization servers), the Windows Server 2012 operating system runs exclusively in a partition known as the Management OS. Created virtual machines run in their own isolated partitions called Guests. Hyper-V hosts can house multiple Hyper-V guest VMs which are isolated from each other in their own partitions. Guest partitions do not have access to the hardware resources. They have a virtual view of the hardware resources known as virtual devices.

Note: The Management OS was formally known as the parent or root partition in Windows Server 2008. These terms are no longer used with Windows Server 2012 Hyper-V.

3.1 Management OS virtual stack components

The Management OS is the primary Windows Server 2012 partition on the host. One of the key components of the Management OS is the virtualization stack. The virtualization stack is the collection of resources that, combined with the Hypervisor, provide the majority of the Hyper-V environment and functionality. The following table details the primary task of each component in the virtual stack.

Table 2 Management OS Virtual Stack Components

Virtual Stack Component	Primary Task
Virtual Machine Management Service (VMMS)	The VMMS is the core of the Hyper-V host. The VMMS manages the state of virtual machines running in the guest partitions (active, offline, stopped, etc.) and controls the tasks that can be performed on a VM based on a current state such as the addition and removal of devices. When a VM is started, the VMMS process is responsible for creating a corresponding Virtual Machine Worker Process. This VMMS can be identified as "VMMS.exe" in get-process PowerShell output. The Hyper-V-VMMS logs in the event viewer are a great place to start troubleshooting the host. The VMMS process runs as VMMS.exe on the Hyper-V host.
Virtual Machine Worker Process (VMWP)	Virtual Machine Worker Processes (VMWP) are started by the VMMS when virtual machines are started. A virtual machine worker process (identified as vmwp.exe) is created for each Hyper-V guest and is responsible for much of the management level interaction between the Management OS and the guest virtual machine. The duties of the VMWP include creating, configuring, running, pausing, resuming, saving, and resorting of the associated guest vm. It also handles IRQs, memory, and I/O port mapping through a "Virtual Motherboard" (VMB) which instantiates all virtual devices (VDevs). Each VMWP has one virtual motherboard.
Virtual Devices (VDevs)	Virtual Devices are managed by the Virtual Motherboard (VMB). a Virtual device is a software module that provides an I/O path for a partition. A VDev allows a single physical device attached to the Management OS to be shared across multiple guest partitions. Each partition believes it has exclusive access to the device. Virtual devices are often packaged as a COM device and managed using a WMI interface. There are two different kinds of Virtual Devices in general: Emulated devices also sometimes called Core devices; and Synthetic devices also called Enlightened devices
Synthetic Device	Synthetic devices (Enlightened Devices) are high performance software drivers that control access to physical hardware devices that are designed for optimal performance in virtualized environments. They are able to leverage more efficient communications mechanisms between virtual hardware and physical hardware – the VMBus

Virtual Stack Component	Primary Task
Emulated Device	Emulated hardware is a software construct that the hypervisor presents to the virtual machine as though it were actual hardware. The drawback is that emulated hardware can be computationally expensive and therefore slow to operate. The software component is a complete representation of a hardware device, which includes the need for IRQ and memory I/O operations. An emulated device requires many Hypervisor intercepts for every single I/O making it very expensive from a performance perspective. Emulation occurs inside Virtual Machine Worker Process (vmwp.exe) for the emulated guest.
Windows Management Instrumentation Provider (WMI)	The WMI provider provides a gateway to the VMMS. WMI is used by tools such as Hyper-V Manager and agents such as those used in Microsoft System Center. The WMI provider also provides a library of functions for developers and scripters to quickly build customer tools, utilities, and enhancements to the virtualization platform.
Virtual Infrastructure Driver (VID)	Operates in kernel mode and provides partition, memory, and processor management for the guest virtual machines. The Virtual Infrastructure Driver (Vid.sys) also provides the conduit for the components higher up the Virtualization Stack to communicate with the hypervisor

3.2 Kernel Mode Hyper-V components for Management OS and guests

Other than the virtualization stack, the Management OS partition also includes other key components which run in kernel mode (ring 0).

Table 3 Management OS partition key components

Component	Location	Primary Task
Virtualization Service Providers (VSP)	Management OS	The VSP resides in the Management OS and provides synthetic device support via the VMBus to the Virtual Service Clients (VSC) running in the enlightened guest partitions. Its primary task is to handle the device access requests from guest partitions. The four types of VSPs are: network, storage, video and human interface devices
Virtualization Service Clients (VSCs)	Enlightened Hyper-V guests	VSCs are synthetic device instances on the guest partitions. They communicate with the corresponding VSPs in the Management OS over the VMBus to satisfy a guest partitions device I/O requests.
Microsoft Virtual Machine Bus (VMbus)	Management OS and enlightened guests	VMBus is a high-speed communication channel between virtual machine VSCs and Management OS VSP. It uses shared memory buffers and memory descriptors to move data between the Management OS and the virtual machine. Shared memory access across partitions is extremely fast, especially as system architecture and hardware technologies have improved over recent years.
Windows Hypervisor Interface Library (WinHv)	Management OS and enlightened guests	The WinHv.sys DLL is located in the Management OS and any Hyper-V enlightened guest. It is a bridge between partitioned operating system drivers and the hypervisor which allows drivers to call the hypervisor using standard Windows calling conventions.
Hypercall	Management OS and enlightened guests	Hypercall is the interface for communication with the hypervisor.

Figure 1 details the various components in the Hyper-V architecture.

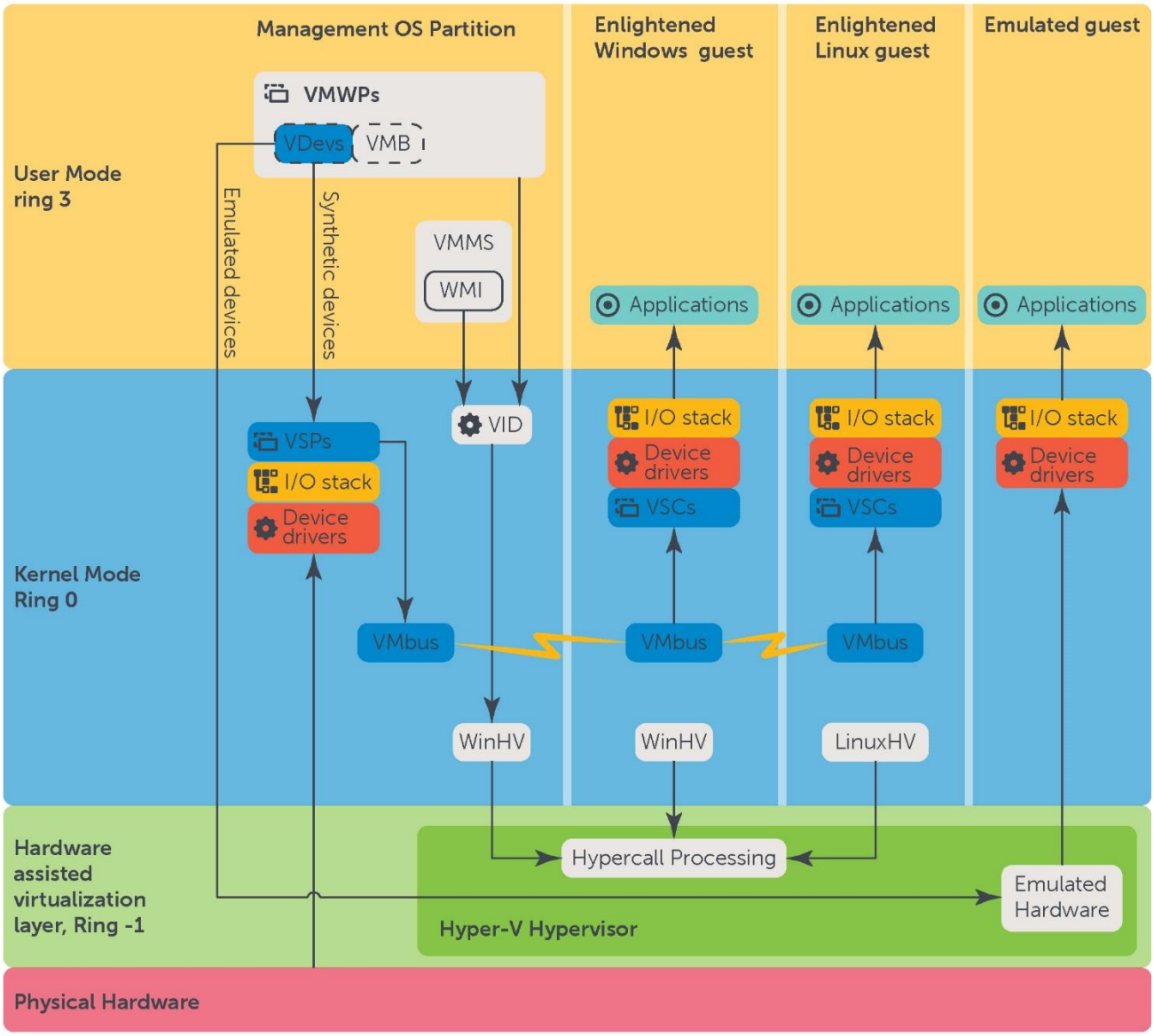

Figure 1 Hyper-V architecture components

Note: In Figure 1, guest partitions do not have direct access to hardware resources. Instead, guest partitions are presented a virtual view of the resources, known as virtual devices (VDevs). Requests to the virtual devices are redirected either through the virtual machine bus (VMBus) or the hypervisor to the management OS, which handles the device requests. The VMBus acts as a logical inter-partition communication channel, with separate channels allocated for communication between the management OS partition and a guest partition.

3.3 Hyper-V guest virtual machines

Windows Server 2012 Hyper-V currently supports the three types of guest virtual machines described in the remainder of this section.

3.3.1 Enlightened Windows guests

This is a virtual machine with a Windows guest OS that has all of the Hyper-V integration components installed (VSC, VMBus and WinHV). Windows Server 2012 and Windows Server 2008 R2 are Windows operating systems that are enlightened for Hyper-V virtual machines. When a guest has these components, it is labeled, "enlightened". In the context of Hyper-V, the term enlightened refers to a guest OS that is aware of and is optimized for running in a Hyper-V virtual environment. Enlightened Windows guests can take advantage of Synthetic Devices in the Management OS. The following table lists enlightened VM hardware components and their corresponding synthetic devices.

Table 4 Enlightened VM hardware and synthetic devices

Synthetic device	Associated virtual machine ardware
Microsoft VMBus network adaptor	This network adapter is added when the integration services are installed into the OS in the virtual machine and the virtual network adaptor is added to the virtual machine.
Microsoft synthetic SCSI controller	This SCSI adapter is added when the integration services are installed into the OS in the virtual machine and a SCSI adaptor is added to the virtual machine. It supports up to 64 devices per controller (max of 4 SCSI controllers per virtual machine).
Pass through storage	Pass through storage provides mechanisms to virtual machines that read and write directly to a storage device presented to the virtual machine. The virtual machine sees it as a disk.
Microsoft synthetic video	This synthetic video adapter is added when the integration services are installed into the OS in the virtual machine.
Microsoft synthetic mouse	This synthetic mouse is added when the integration services are installed into the OS in the virtual machine.
Microsoft Hyper-V synthetic virtual Fibre Channel adapter	This synthetic virtual Fibre Channel adapter is added to the guest when the integration services are installed into the OS in the virtual machine and a virtual HBA is added to the virtual machine.
Microsoft Hyper-V virtual PCI Bus	This synthetic virtual PCI Bus is added when the integration services are installed into the OS in the virtual machine and SR-IOV networking is being used.
Microsoft Hyper-V generation counter device	This synthetic virtual machine generation counter device is added when the integration services are installed into the OS. The device provides a 128-bit VM generation identifier for a VM that stays the same unless the VM is reverted to a snapshot.

Synthetic devices can be thought of as proxy devices that present themselves as a real device, but only serve to pass data along the VMBus from the VSP on the Management OS to the VSC on the enlightened guest partitions. This process does not require software emulation, and therefore offers higher performance for virtual machines and lower host system overhead.

3.3.2 Enlightened Linux Guests

Contrary to popular belief, Microsoft Server 2012 Hyper-V does support several Linux distributions for use on Hyper-V guest virtual machines. In fact, if the Linux Virtual Integration Services (VIS) is installed, these Linux guests can be fully enlightened. The Linux VIS is a set of drivers that enable synthetic device support for supported Linux distributions. When installed in a supported Linux virtual machine running on Hyper-V, the Linux Integration Components provide:

- Driver support: Linux Integration Services supports the network controller and the IDE and SCSI storage controllers that were developed specifically for Hyper-V.
- Fastpath Boot support for Hyper-V: Boot devices now take advantage of the block Virtualization Service Client (VSC) to provide enhanced performance.
- Time Sync: The clock inside the virtual machine will remain synchronized with the clock on the virtualization server with the help of the pluggable time source device.
- Integrated Shutdown: Virtual machines running Linux can be shut down from either Hyper-V Manager or System Center Virtual Machine Manager by using the **Shut down** command.
- Symmetric Multi-Processing (SMP) support: Supported Linux distributions can use multiple virtual processors per virtual machine. The actual number of virtual processors that can be allocated to a virtual machine is only limited by the underlying hypervisor.
- Heartbeat: This feature allows the virtualization server to detect whether the virtual machine is running and responsive.
- KVP (Key Value Pair) exchange: Information about the running Linux virtual machine can be obtained by using the Key Value Pair exchange functionality on the Windows Server 2012 virtualization server.
- Integrated Mouse Support: Linux Integration Services provides full mouse support for Linux guest virtual machines.

As of the time of this writing, the latest version of the Linux VIS from Microsoft is V3.4. The supported Linux guest operating systems are: Red Hat Enterprise Linux 5.7, 5.8, 6.0-6.3 x86 and x64 and CentOS 5.7, 5.8, 6.0-6.3 x86 and x64.

3.3.3 Emulated Guests

Emulated guests are guests that are not Hyper-V aware. This means that they cannot take advantage of Synthetic Devices or other Hyper-V optimizations. Emulated guest utilize emulated devices. This software component implements a unified least-common-denominator set of instructions that are universal to all devices of a particular type (such as IDE drive controllers, VGA Graphics Cards and legacy network adapters). This all but guarantees that it will be usable by almost any operating system, even those that Hyper-V Server does not directly support. A typical use case for Emulated guests occurs when there is a legacy application running on an older, unsupported operating system such as Windows NT or Windows Server 2000. As an example, an organization might wish to upgrade the version of Windows Server but cannot because the application vendor is no longer in business and there are no updated drivers for newer hardware or operating systems. These legacy systems and applications are good candidates to run as emulated guests because Hyper-V would abstract the hardware using device emulation. Even if there is a decrease in performance, the application will still be operational until a replacement is found.

Note: This paper discusses Hyper-V emulated devices and emulated guests for completeness. Going forward, this paper will not discuss device emulation and emulated guest virtual machines. The remainder of this paper is focused on Hyper-V enlightened devices and enlightened guests.

4 Hyper-V networking

4.1 The Hyper-V virtual switch

Some of the biggest improvements in Windows Server 2012 Hyper-V are related to Networking. Windows Server 2012 Hyper-V introduces a manageable layer 2 extensible switch into the product. In fact, the Hyper-V extensible virtual switch is a cornerstone of the Microsoft Server 2012 cloud operating system strategy to move beyond virtualization. The Hyper-V virtual switch

The Hyper-V virtual switch in Windows Server 2012 is a layer-2 virtual network switch that provides programmatically managed and extensible capabilities to connect virtual machines to the physical network. The Hyper-V virtual switch is extensible because as an open platform, it lets multiple vendors provide standard Windows API framework extensions. The reliability of extensions is strengthened through the Windows standard framework and reduction of required third-party code for functions. It is also backed by the Windows Hardware Quality Labs (WHQL) certification program. The Hyper-V Extensible Switch and its extensions can be managed by using Windows PowerShell, or programmatically with WMI or through the Hyper-V Manager user interface.

In simplistic terms, the Hyper-V virtual switch is a virtual representation of a physical NIC on the host. Being able to virtualize the host physical network components allows for the creation of a pool of network resources that can be shared among the guest virtual machines. This is similar to the way it is done with memory and CPU resources. The Hyper-V virtual switch performs the same basic task as a physical switch except that it is connecting the virtual network interface controllers (vNICs) of the guest VMs to the network. Like a physical switch, the Hyper-V virtual switch manages the shared pool of network resources by performing the following operations:

- Traffic isolation and flow
  - Port access control lists (ACLs) are used to allow or deny specific addresses to move through the network.
  - Private virtual LANs (PVLANs) let IT administrators establish a gateway without the need to define a strong two-tier network.
  - Trunk mode in Windows Server 2012 Hyper-V allows multiple VLANs to be used on a virtual machine network adapter. Previously, when Hyper-V sent VLAN traffic to a virtual machine, it could only choose a single VLAN per virtual machine.
- Traffic shaping
  - Quality of Service (QoS) is used to set minimum and maximum bandwidth levels by using absolute or relative amounts. QoS can be used to guarantee minimum levels of bandwidth so that service level agreements are met, and also to minimize or prevent excessive usage by specific clients.
- Security
  - Dynamic Host Configuration Protocol (DHCP) guard is used to control whether or not a virtual machine is allowed to behave as a DHCP server, which can help prevent network attacks involving the deliberate misuse of addresses.

  - IP security (IPsec) task offloads enables virtual machines to offload IPsec encryption directly to the IPsec offload engine on a network adapter.
- Performance Enhancements
  - Dynamic virtual machine queues (VMQs) are supported by Windows Server 2012to adjust the number of cores used by the host virtual switch base on traffic load.
  - Single Root I/O Virtualization (SR-IOV) accelerates performance by letting network traffic go directly to a virtual machine.
- Diagnostics
  - Port mirroring provides the ability to copy traffic from multiple virtual machines to multiple port switches to help identify network issues.
  - Event Tracing for Windows (ETW) helps IT managers to easily diagnose issues with a switch and related extensions without having to use a debugger.

As stated earlier, guest partitions do not have direct access to hardware resources. For network access to guest partitions, a Network VSC (NetVSC) runs in a guest operating system. Networking requests and packets are sent between each NetVSC and the Network VSP that runs in the management OS. The NetVSC also exposes a virtualized view of the physical network adapter on the host computer. This virtualized network adapter (vNIC) is known as a synthetic network adapter.

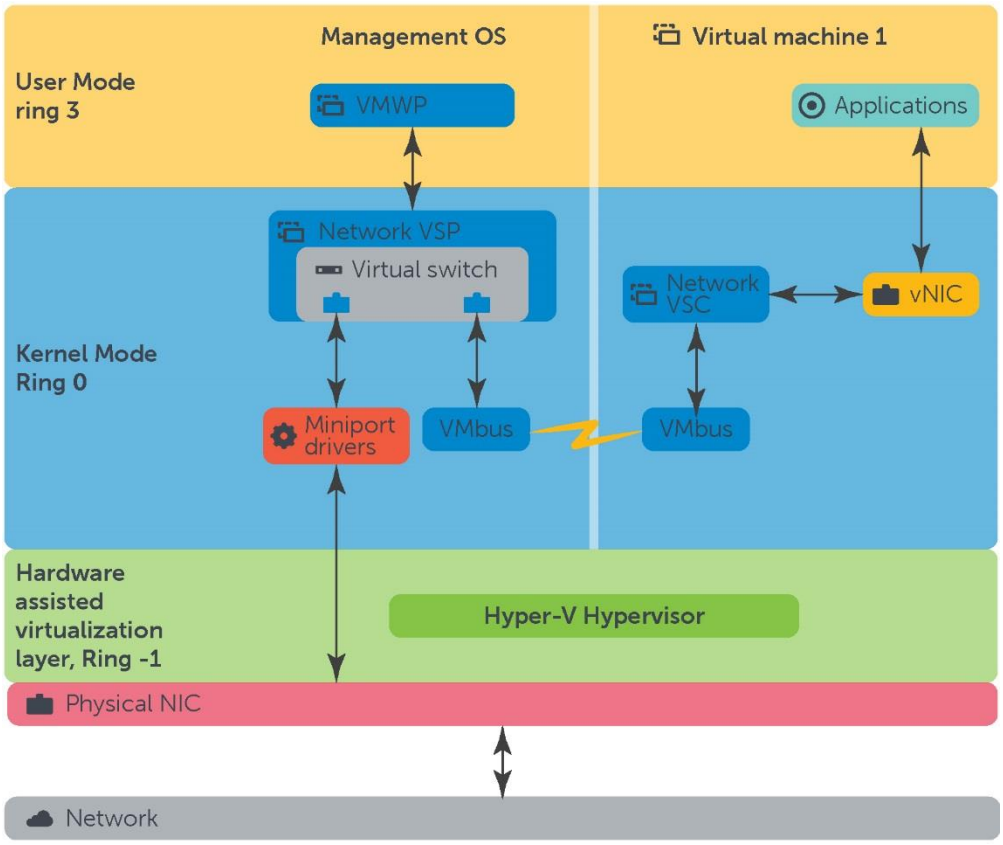

Figure 2 Hyper-V virtual switch traffic and data path

Applications running on the guest VMs can access network resources through the vNICs. These vNICs are presented to the guest OS by the Network VSC, that is being presented a synthetic virtual switch device through the associated Network VSP running on the Management OS. The network VSC forwards packets to and from the associated virtual switch port over the VMBus to the Network VSP switch driver.

An important aspect of virtual switches is that they are unique to a host and are not distributed. This can create some issues when migrating Hyper-V VMs between hosts. Failover Clustering, which is discussed later in this document, can overcome these issues.

Hyper-V has three types of virtual switches: external, private and internal.

External virtual switch

An external virtual switch is used to connect each vNIC to a physical network and represent a single physical connection. Each vNIC has its own unique MAC address and a unique IP address. Each vNIC is completely separate from the other vNICs and from NICs that are being used by the Management OS in the host itself.

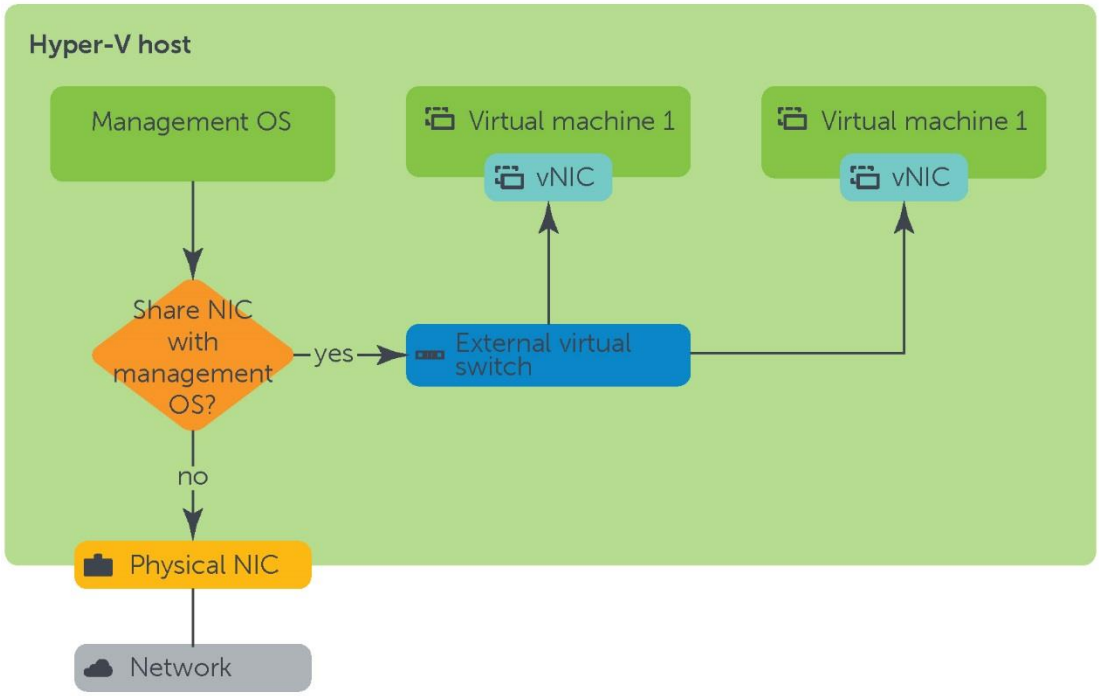

Figure 3 External virtual switch configuration

Virtual NICs are not just confined to the virtual machines on the host. The Management OS can utilize vNICs, but some network administrators prefer to isolate the Management OS from the virtual machine traffic. This is decided when creating the external virtual switch. If **Allow Management Operating System to Share This Network Adapter** is selected when creating the external virtual switch, both Management OS and virtual machine network traffic will use the same external virtual switch. This option is selected by

default so that the Management OS and VMs share the same external virtual switch. Deselect this option to isolate the Management OS and VM network traffic from each other.

Private virtual switch

The private virtual switch is not connected to either the management OS or the physical network. It is completely isolated, and therefore any vNICs that are connected to the private virtual switch are isolated too. A typical use case for the private virtual switch is to secure sensitive workloads that should not be open to contact by the physical network or other virtual machines. An example of this is one virtual machine that is running a firewall service and has two vNICs: one that is connected to an external switch, and the other connected to the private network for the sensitive workload. The firewall would then control and route access to the private network. This configuration is shown in the following diagram.

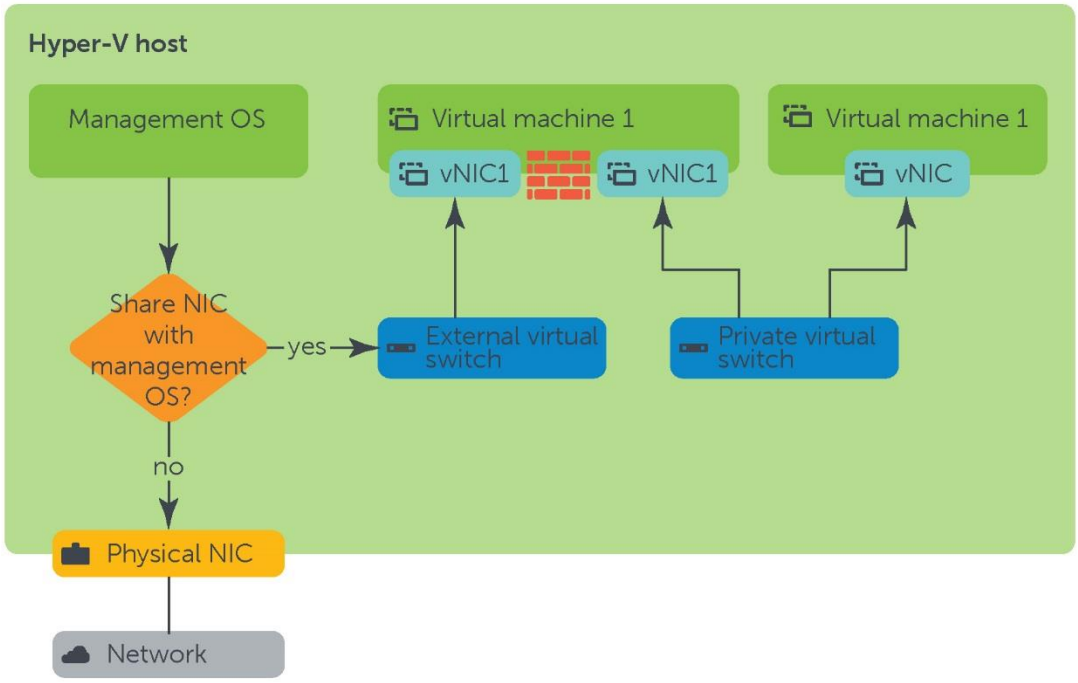

Figure 4 Private virtual switch configuration

Internal virtual Switch

An internal virtual switch is essentially the same as a private virtual switch, except that it allows a vNIC connection from the Management OS. Like the private virtual switch, it has no connection to the physical network, thereby limiting its usefulness in production environments. The internal virtual switch is most often used for live migrations or heartbeat type networks between clustered Hyper-V guest virtual machines on the same host.

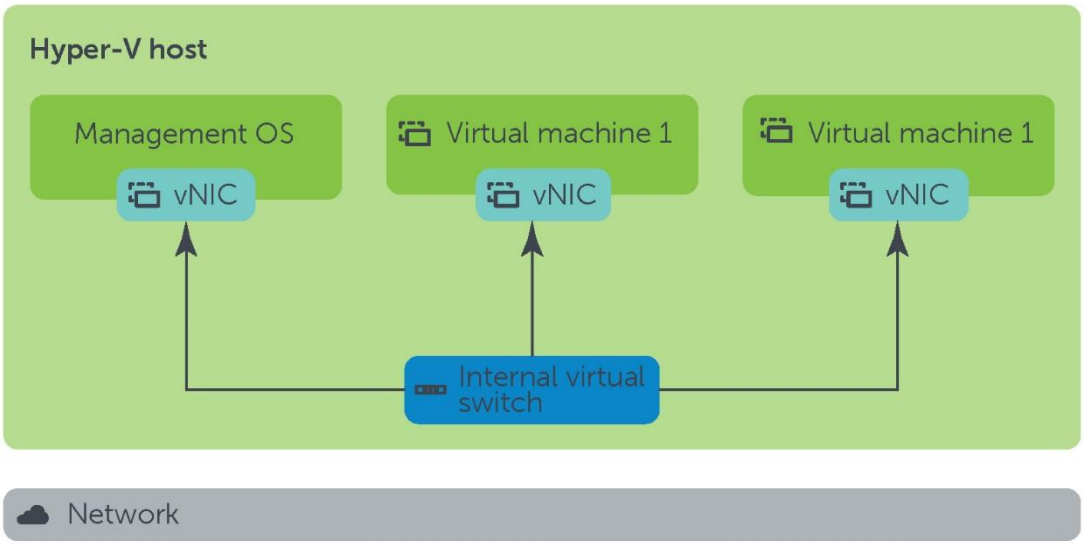

Figure 5 Internal virtual switch configuration

The following table summarizes the three types of Hyper-V virtual switches.

Table 5 Hyper-V virtual switch type summary

Virtual Switch Type	Can connect to external network?	Can be Used by Management OS?	Primary Use Case
External Virtual Switch	Yes	Yes	All Workloads
Internal Virtual Switch	No	No	Secure Workloads
Private Virtual Switch	No	Yes	Lab / Live Migrations / Heartbeat

4.1.1 Virtual LANs (VLANs)

Simply put, a VLAN is a group of network hosts that behave as if they shared a common network segment, even though the hosts might all be physically connected to different segments. What VLANs provide is a method of dividing a physical LAN into multiple subnets. This is done because VLANs can create more address space, control broadcast domains and isolate network traffic.

For Hyper-V, VLANs are most commonly seen in larger environments where live migrations of VMs can take place between hosts on different subnets. Placing each virtual machine onto a common VLAN ensures that the VM can still communicate over the network, even though it might have been moved to a host on a different subnet. VLANs for Hyper-V are only used when the Hyper-V hosts are part of a large multi-site cluster where live migrations occur. For most SMB environments that have localized failover clusters with hosts on the same subnet and a few dozen Hyper-V VMs that participate in live migrations, VLANs could introduce unwanted complexity.

4.1.2 NIC teaming

NIC teaming, also known as Load Balancing/Failover (LBFO) enables multiple NICs to be placed in a team interface. This provides bandwidth aggregation and traffic failover, which prevents loss of connectivity in the event of a NIC failure on the host. In Windows 2012, a NIC team can be made up of one to 32 physical NICs. These teamed NICs are known as team members. Team members can be from different manufactures and even support different network speeds (although not recommended). The only requirement for NIC teaming is that a team member NIC be on the Windows Server 2012 hardware compatibility list (HCL), which can be found at www.windowsservercatalog.com. A NIC team can be utilized to create a Hyper-V virtual switch.

NIC teaming can be used for NICs processing regular network traffic but not for iSCSI traffic (iSCSI does not support NIC teaming and requires Multipath I/O (MPIO) for LBFO). Be sure iSCSI network traffic has two or more dedicated NICs, each with their own virtual switch.

4.1.3 Receive Side Scaling and Dynamic Virtual Machine Queuing

Receive Side Scaling (RSS) plays an important role in Windows Server 2012 overall networking strategy. Traditionally, a single core in the processor has been used when packets arrive in a NIC. As servers and workloads have grown, this single core solution has become a resource constraint. The introduction of 10Gb networks as aggravated this issue even more. Essentially, a the processing power of single core can no longer keep up with growing network traffic demands from modern servers. RSS uses a series of queues on the NIC that allow operating systems to dynamically scale processing of non-virtualized network traffic across multiple cores in the server. Even though RSS works on non-virtualized workloads, it is important for Hyper-V because RSS is an enabler for SMB Multichannel on high capacity NICs (10Gb) which results in faster Cluster Shared Volume (CSV) storage operations. RSS requires RSS enabled NICs and will yield best results on 10Gb+ networks.

Dynamic Virtual Machine Queuing (dVMQ) provides the same functionality for virtualization traffic passing through a virtual switch that RSS does for non-virtualized traffic. dVMQ allows for vNIC traffic being processed by a virtual switch to use more than one core and uses the same queues on the NIC that RSS does. NIC queues can be dedicated to RSS or dVMQ, but not both. This means that if the NIC is going to be used in a NIC team, the team needs to be configured for either RSS or dVMQ. Most modern 10Gb NICs support dVMQ; it is enabled by default when your create a vNIC in Hyper-V.

Note: RSS and dVMQ can introduce significant complexity into a virtualized environment. Tuning the parameters for these features require a solid understanding and upfront planning that are not for the novice Hyper-V administrator. Also, the performance improvements that RSS and dVMQ can provide are not noticeable unless there is a high amount of 10Gb network traffic coming into a server.

4.1.4 Single root I/O virtualization

Guest virtual machines share a pool of virtualized network hardware resources for their I/O path. With current Hyper-V networking virtualization, this I/O path resource sharing is managed by the hypervisor and management OS that consumes host CPU cycles. For most virtual machine environments and virtual applications, the CPU cycles consumed by the OS and hypervisor resource management do not result in performance detriments. However, as virtual machines grow and scale in number, the demands on the host CPU resource pool grow as well. It is conceivable that as virtualized environments scale up and out, it is inevitable that the CPU cycles consumed by host based resource management will result in CPU resource constraints.

Windows Server 2012 Hyper-V uses single root I/O virtualization (SR-IOV) to offload the management for the network I/O path virtualization between a network, physical host network cards, and virtual machines directly to the network card hardware itself. SR-IOV is an extension of the PCI Express (PCIe) standard that allows PCIe device resources to be shared among multiple virtual machines by providing them a direct hardware path for I/O. The management of these PCIe resources is performed by the SR-IOV capable network card, bypassing the hypervisor and management OS, and freeing up the CPU resources that would otherwise be involved.

SR-IOV allows a device, such as a network adapter, to separate access to its resources among physical and virtual PCIe hardware functions.

Physical Function (PF): The PF is he primary function of the device that advertises the device SR-IOV capabilities. The PF is associated with the Hyper-V Management OS partition in a virtualized environment.

Virtual Functions (VFs): The VFs are the virtual interfaces on a PCIe NIC where hardware resources are partitioned using SR-IOV capable NICs. They allow the adapter resources to be shared in a virtual environment.

Each VF is associated with the device PF. A VF shares one or more physical resources of the device, such as a memory and a network port, with the PF and other VFs on the device. Each VF is associated with a Hyper-V guest partition in the virtualized environment. This allows the network traffic to completely bypass the virtual switch infrastructure in the management OS, greatly increasing throughput and reducing required host CPU cycles required for resource management.

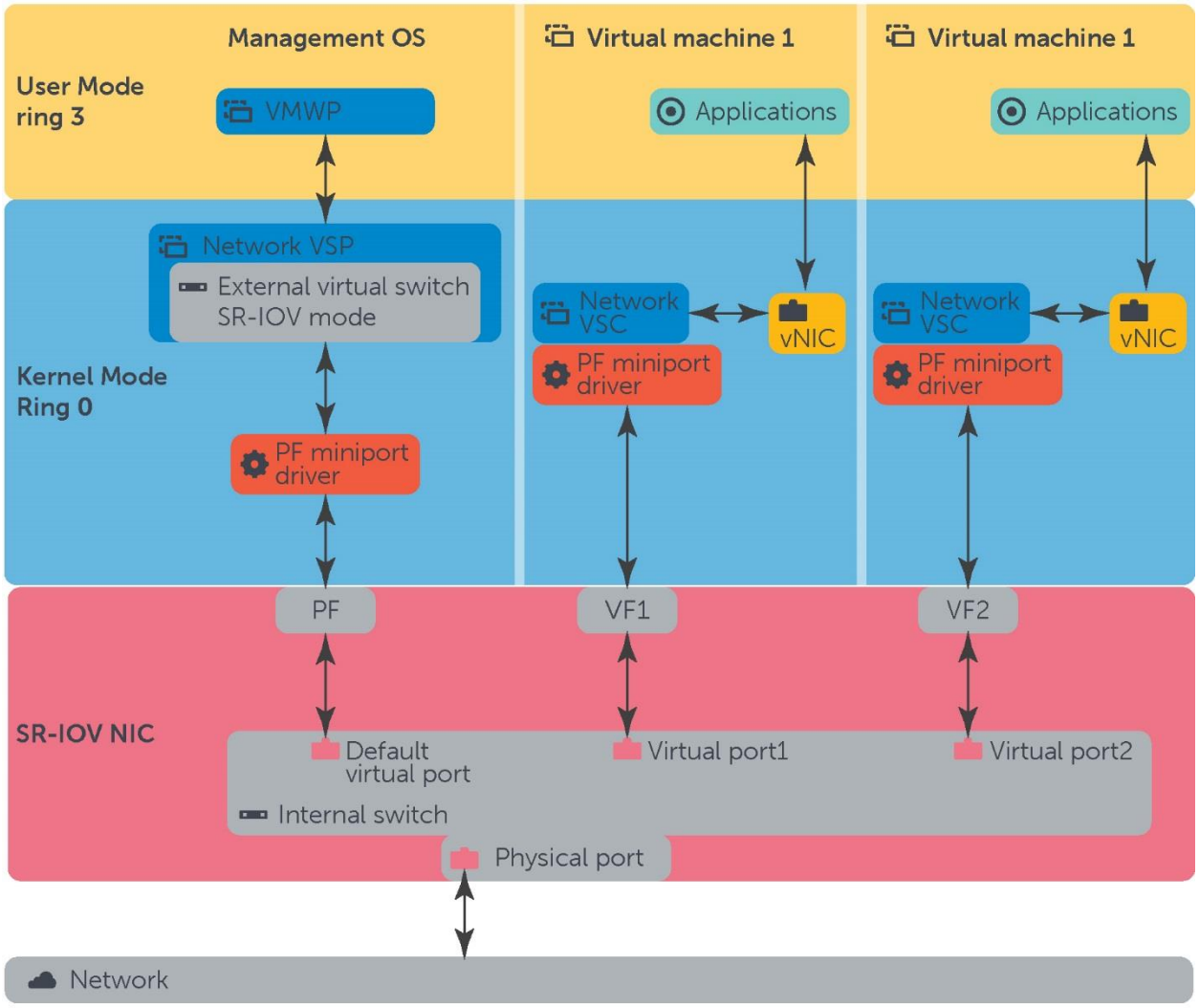

Figure 6 Data path within the SR-IOV interface

Figure 6 shows that all data flows directly between the protocol stacks in the guest virtual machine system and the VF on the SR-IOV NIC. This eliminates the overhead of the typical synthetic data path where data packets flow between Hyper-V guests and management OS partitions.

Note: SR-IOV cards cannot be used in NIC teaming.

Some of the other key components shown in the diagram are:

NIC Switch: A hardware component of the NIC that supports the SR-IOV interface. The NIC switch forwards network traffic between the physical port on the adapter and internal virtual ports (VPorts). Each VPort is attached to either the PF or a VF.

PF miniport driver: This driver is responsible for managing resources on the network adapter that are used by one or more VFs. Because of this, the PF miniport driver is loaded in the management operating system

before any resources are allocated for a VF. The PF miniport driver is halted after all resources that were allocated for VFs are freed.

Physical Port: A hardware component of the network adapter that supports the SR-IOV interface. The physical port provides the interface on the adapter to the external network.

VF miniport driver: This driver is installed in the VM to manage the VF. Any operation that is performed by the VF miniport driver must not affect any other VF or the PF on the same network adapter.

Virtual Ports (VPorts): A data object that represents an internal port on the NIC switch of a network adapter that supports the SR-IOV interface. Similar to a port on a physical switch, a VPort on the NIC switch delivers packets to and from a PF or VF where the port is attached.

In theory, SR-IOV can offer the following to virtualized environments:

- Scalability for the future.Each SR-IOV NIC can have many VFs (up to 256 depending on the hardware). This means up to 256 vNICs could be connected directly to the physical SR-IOV NIC
- Reduced network latency for virtualized workloads.
- Increased throughput.
- Lower host resource utilization..
- Increased levels of security. VFs are designed to channel data and have no accessible configuration options.

4.1.5 Quality of Service and bandwidth management

As stated earlier, network virtualization results in a pool of network resources which can be shared among virtual machines. The reality is that in production environments, not all virtual machines are equal. Some virtual machines will be running heavier workloads which will demand more resources than other less loaded virtual machines. With the Quality of Service (QoS) enhancements found in Windows Server 2012, an administrator can now guarantee a bandwidth service level to a particular Hyper-V virtual machine using a combination of primary and secondary rules.

4.1.5.1 Primary QoS Rules

Absolute, bits per second (bps) based rules: The bps-based rules can guarantee a certain amount of bandwidth for a particular virtual machine or application. For example, if there is a total of 10Gbps bandwidth available, you could create a rule that specifies a particular virtual machine would be guaranteed three Gbps at any time. An advantage of bps rules is that they are very easy to communicate to customers who might be sharing a particular network pipe. QoS bps rules guarantee with certainty a particular bandwidth to customer VMs. However, they can be inflexible, especially if workloads need to be mobile between hosts. In a failover situation, a virtual machine might be moved to a host which has only a one Gbps NIC. Because the bps rule is applied to the vNIC of the VM, attempting to move the VM to the new host will cause issues.

Share of bandwidth, weight based rules: Weight based QoS rules offer more flexibility than bps rules because there is no consideration of the actual link speed in bps. Using the scenario in the previous paragraph, if a weight based rule for 30% of guaranteed bandwidth was applied to the VM on the 10Gbps

NIC host, then moving it to another host with a 1 Gbps NIC would not be an issue. In this case, the VM is guaranteed 30% of the available bandwidth or 300Mbps of an available 1 Gbps bandwidth.

4.1.5.2 Secondary Rules

Minimum bandwidth: Minimum bandwidth guarantees a minimum of a host available bandwidth. This can be viewed as a bandwidth floor. For example, if a VM was guaranteed a minimum of 25% of a host bandwidth, then the VM would always have 25% of the total bandwidth as a minimum. If the VM wasn't using 25% of the bandwidth, this unused bandwidth would be available to other VMs running on the host. Minimum bandwidth rules guarantee a Service Level Agreement (SLA). Combining minimum bandwidth and weight based rules is considered the most flexible of all the QoS configurations for Hyper-V.

Maximum bandwidth: Maximum bandwidth is a cap on the amount of bandwidth a virtual machine or service can access. Placing a cap on bandwidth access is quite useful when an administrator wants to cap the available bandwidth for client virtual machines. Maximum bandwidth rules can be applied to certain VMs notorious for periodic heavy burst workloads in an effort to prevent network congestion. These rules are designed to limit bandwidth, but in turn, they also limit flexibility.

Note: The focus of Windows Server 2012 Hyper-V QoS was to increase flexibility. Therefore implementing a combination of weight based and minimum bandwidth rules is considered the most flexible solution for Hyper-V QOS because it makes no assumptions about hardware capacity and is elastic in nature. This allows virtual machines to burst beyond minimum values when needed.

4.1.6 Advanced Protocols - Data Center Bridging and RDMA

Data Center Bridging (DCB) is an extension to the Ethernet protocol that allows QoS rules to be applied at the hardware level. This results in a decreased CPU load on the server. DCB must be end-to-end network support in order to function. All of the host NICs, switches, and storage arrays on the network must support the protocol. Remote Direct Memory Access (RDMA) is the primary enabler for Server Message Block (SMB) Multichannel. This protocol allows for the use of 100% of bandwidth in high bandwidth networks without using any CPU processing power on the server. SMB Direct accomplishes this by offloading the SMB transfer so that it is invisible to the operating system. DCB and RDMA are technologies most often seen in environments which require very high throughput loads. These loads are often well out of the range seen by the majority of small and medium businesses (SMB).

4.2 Hyper-V network design considerations for SMB

There is more than one way to correctly design a Hyper-V network. However, some designs are better suited for specific networks. The design process is a matter of balancing current and future network workload requirements with the appropriate amount of controls and complexity.

Core objectives that a Hyper-V network design should meet are listed in the remainder of this section.

4.2.1 Reduce the amount of deployed NICs in the network

For all networks, whether for SMB or the largest multinational corporation, a fundamental design principle is to reduce the amount if NICs and switch ports deployed in the network. Among other reasons, this has two extremely desirable results.

Simpler data center management:

- Faster configuration and deployment of new server racks
- Easier management for administrators
- Fewer cables for a more efficient rack airflow, which further reduces cooling requirements
- Increased data center reliability due to fewer connections. There is also a decreased risk for the wrong cables getting pulled, and there are fewer NICs and switch ports that can go bad.

Lower Energy Costs

Using fewer deployed NICs means fewer deployed switches which can result in lower cooling infrastructure costs. According to a study in Computer World from April 2006, the average NIC consumes 12.5 Watts, and up to an additional 0.8 to 1.5 Watts/Watt consumed by electronics is required for cooling. This means that if the workloads of two NICs could converge into a single NIC, the server not only conserves 12.5 Watts for the NIC, it also conserves up to an additional 18.75 Watts for cooling. This results in total energy consumption decrease of over 31 Watts per converged NIC. If an environment is able to converge two NICs per server in a ten-server rack, then there is a 620 Watt decrease in energy consumption (31 W x 2 x10 servers) for the servers in the rack alone. When the power savings is factored into the decreased amount of switch ports needed to support the rack, the power savings could easily approach 1 KW per rack. Considering that the average datacenter rack consumes 5 KW of power, 1 KW represents a 20% energy consumption reduction per server rack in the data center.

In summary, combining different network workloads through a single NIC results in fewer deployed NICs, fewer deployed switches, fewer cabling issues, easier management, and lower energy consumption. These benefits are a primary design intent when designing a converged fabric.

4.2.2 Select appropriate Hyper-V network technologies

Although Hyper-V networking offers a wide variety of features and controls, many of these features are targeted for enterprise scale, geographically dispersed, and public cloud environments. While these features are powerful, they can add considerable complexity into a SMB environment, with little realized overall gain. With relation to SMBs, follow the keep–it-simple philosophy when designing the converged network for a Hyper-V environment. The network engineer designing a SMB converged network (data and LAN traffic combined) should focus on using networking virtualization technologies which offer the most bang for the buck while adding minimal complexity.

Table 6 Key networking features and their introduced complexity with respect to Small Medium Businesses

Hyper-V networking technology	Required technology for MSCF*	Acquisition cost	Introduced complexity
Hyper-V Virtual Switch	Yes	Free	Varying
Virtual LANs	No	Free	Medium
NIC Teaming**	No	Free	Medium
RSS and DVMQ	No	Medium	Large
SR-IOV	No	Medium	Small
QOS	No	Free	Medium
DCB and RDMA	No	Large	Large

***** MSCF: Microsoft Converged Fabrics

****** Technology not supported for all networking protocols (i.e. iSCSI) or networking technologies (SR-IOV)

Table 6 shows that NIC Teaming, QoS, and SR-IOV are technologies that achieve a relatively strong return with minimal complexity for SMB Hyper-V networks. The other technologies are best suited for enterprise class data centers with high network workloads where the higher acquisition costs and introduced complexity are offset by the increased performance and scalability.

4.2.3 Know the current workloads and properly scale for future growth

High-performance applications require more bandwidth, which makes investment in higher-speed networks (10GbE) cost effective. Higher speed networks require fewer NICs and switch ports, thereby lowering management and energy costs. In addition, increased network bandwidth and utilization will allow administrators to migrate application data and virtual machines from one environment to another with minimal impact to production.

The design plan for Hyper-V SMB networks must include existing workloads with existing consumed bandwidth plus future expansion. The current proliferation in 10 GbE networks is primarily due to 10GbE being the default installed configuration on new network and server hardware and not necessarily being driven by expanding network workload requirements. Many workloads in the SMB space have not begun to tax existing 1 GbE infrastructures.

4.2.4 Select the appropriate storage interface option

A typical converged networked environment consists of application servers, external hardware interfaces within the application servers, cabling, and a storage area network (SAN) between the servers and arrays.

The external interface technologies, as components of these converged environments, are the foundation of the overall storage framework's performance, scalability, reliability, technical complexity, and cost. Several interface options have been proposed to support converged environments. The primary Hyper-V storage interfaces considered today are:

- Serial Attached SCSI (SAS)
- Fibre Channel (FC) protocol
- Internet Protocol SCSI (iSCSI)

With this range of interface options, each with its own distinct features and characteristics, it is important to examine the strengths, position, and special considerations of each one.

Table 7 Interface option details

	iSCSI	**SAS**	**FC**
Description	Interconnect technology built on SCSI and TCP/IP	Serial protocol for data transfer incorporating SCSI command	Transporting SCSI command sets, in the case of disk arrays.
Architecture	IP-based standard—SCSI commands send in TCP/IP packets over Ethernet	Serial, point-to-point with discrete signal paths	Switched—multiple concurrent transactions
Distance between disk array and to a node (server or switch)	Unlimited, however, latencies increase as distances increase	8 meters between devices, 25 meters with use of SAS Switches	50 Km standard, up to 100 Km with distance extension technology
Scalability	No limits to the number of devices (subject to vendor limitations)	Up to 1000 devices with use of SAS Switch (including servers and drives)	256 devices per FC target 16 million SAN devices with the use of switched fabric
Performance	10 Gbps 40 Gbps standard just released – but little hardware available yet 100 Gbps standard on horizon	6 Gbps with x4 wide ports for theoretical bandwidth up to 24 Gbps 12 Gbps Standard on horizon	Up to 16 Gbps with 8 Gbps commonly deployed 32 Gbps standard on horizon
Hyper-V integration	iSCSI devices can be presented directly to a virtual machine using various Hyper-V networking technologies	SAS Storage can be presented to VMs as pass through devices	FC SAN storage can be presented directly to a VM using Hyper-V virtual Fibre Channel and virtual HBAs

	iSCSI	SAS	FC
Replication and Disaster Recovery Technologies	Storage Array Based	Host and Application Based	Storage Array Based
Investment	Low to Medium – can use an existing IP network	Medium – scalability requires the implementation of SAS switch architecture which increases costs	High
IT Expertise Required	Medium – Requires some storage and IP cross-training	Low	High
Strengths Summary	Cost, performance, simplicity, and pervasiveness	Cost, performance, simplicity	High performance, scalability, enterprise class reliability and availability
Weakness Summary	iSCSI Overhead (A non-issue when deployed with 10 GbE networks)	Distance and scalability, Replication relies on host and application based technologies	Cost and Complexity
Additional Notes	Over the decade, iSCSI-networked storage solutions have rapidly entered the IT mainstream, offering a secure, reliable, and flexible network storage solution. iSCSI is an attractive option for organizations where cost and simple management is key	SAS is well suited for the entry-level user who is transitioning from DAS in hopes of transitioning unconsolidated and dispersed storage into a shared environment. The impact and adoption of SAS in networked/shared environments will likely take hold in the coming years.	The well-established FC interface currently dominates the enterprise SAN architecture, providing the performance, distance, and connectivity required for these demanding environments. FC will continue to be the leading interconnect for large SANs due to its robustness, performance, and scalability advantages.
Optimal Environments	SMBs and enterprise, departmental and remote offices • Business applications running on top of smaller Oracle or IBM DB2 databases • All Microsoft Business Applications such as Exchange, SharePoint, SQL Server	Infrastructures within close proximity to all devices • Transaction-sensitive databases • Data streaming • All Microsoft Business Applications such as Exchange, SharePoint, SQL Server	Enterprise with complex SANs: high number of IOPS and throughput • Non-stop corporate backbone including Mainframe • High intensity OLTP transaction processing for Oracle, IBM DB2, Large SQL Server databases • Quick response network for imaging and data warehousing • All Microsoft Business Applications such as Exchange, SharePoint, SQL Server

Why iSCSI is often used in Small Medium Business Hyper-V Environments

Table 7 outlines the strengths and weaknesses of the various storage interface options as a storage interface for SMB Hyper-V networks. Each customer has their own set of unique criteria to use in evaluating different storage interface for their environment. For most SMB environments looking to implement a Hyper-V environment, the determining factors for a storage interface are up-front cost, scalability, Hyper-V integration, availability, performance, and the amount of IT Expertise required to manage the environment. Table 7 clearly shows that iSCSI provides a great blend of these factors.

When price versus performance is compared across all of the technologies, iSCSI is more than competitive with SAS and FC in SMB Hyper-V environments. For the SMB space, many environments are not pushing enough IOPS to saturate even the current one Gbps bandwidth levels. At the time of this writing, 1 Gbps networks are becoming legacy and 10Gbps network are the more commonly deployed and recommend technology. Furthermore, by the end of 2015, plenty of storage, switch, and server manufactures will support 40 GbE. This makes 40 Gbps iSCSI a real option for future growth and scalability.

Another reason that iSCSI is considered an excellent fit for SMB Hyper-V implementations, is that iSCSI uses Ethernet networking and fits in extremely well with a Hyper-V converged network vision. Isolating iSCSI NICs on a Hyper-V host allows each one to have its own virtual switch and specific QoS settings. Hyper-V VMs can be provisioned with iSCSI storage directly through these virtual switches, bypassing the management OS completely and reducing I/O path overhead.

Making a sound investment in the right storage interface for an SMB organization is a critical step in designing a Hyper-V virtualized environment. By selecting iSCSI technology, an SMB can ensure that it can cost effectively meet its data needs.

Special Network Consideration for iSCSI

While iSCSI is an excellent storage interface choice for Hyper-V networks, it has two special considerations when factoring into the network design.

- **Overhead:** iSCSI was created based on the idea that a block level storage technology could co-exist on well-established 1 Gigabit Ethernet TCP/IP networks. The driving force behind this technology was the concept that iSCSI could run entirely on existing commodity networking components such as embedded NICs, CAT- 5/6 cabling, and lower cost Ethernet switches and routers. Because iSCSI was based on this concept, it was architected to reside on top of the TCP and IP layers of the network stack of multiple operating systems. Early iSCSI networking deployments sometimes reported lower than expected IOPS, throughput, and higher than expected CPU utilizations due to the overhead from the iSCSI layer. **The overhead of the iSCSI layer, plus other known inefficiencies with TCP/IP was considered a negative especially in legacy 1 GbE network design; however, with modern 10 GbE networks, the effect of iSCSI overhead has become negligible and is no longer relevant as a networking design consideration.** The effects of iSCSI overhead in legacy networks can be mitigated by implementing more advanced network cards which include technologies such SR-IOV, RSS, and DVMQ, and increasing jumbo frames in size from 1,500 bytes to 9,000 bytes.

- **Security:** The implementation of an ISCSI SAN within an existing network can lead to security vulnerabilities if implemented poorly. A security best practice for an ISCSI SANs is to separate data traffic from the normal LAN traffic by deploying iSCSI data traffic to a physically separate network using separate NICs on the host. Separating the network and iSCSI data traffic also has the additional benefit of reducing the potential for network congestion. iSCSI also inherently provides its own protection. This security is implemented by the Challenge Handshake Authentication Protocol (CHAP). CHAP verifies the identity of iSCSI hosts in which the iSCSI host and the iSCSI storage system share a predefined secret. The iSCSI host is authenticated and the transfer of data can occur if the values match. Further security measures can be taken by implementing Internet Protocol Security (IPSec). Internet Protocol security uses cryptographic security services to protect communications over Internet Protocol (IP) networks. IPsec supports network-level peer authentication, data origin authentication, data integrity, data confidentiality (encryption), and replay protection.

4.3 Hyper-V network design templates for Small Medium Businesses

This section brings the various Hyper-V converged network methodologies together into some basic blue prints for deploying Hyper-V converged networks that include iSCSI storage. The three schematic diagrams are increase in complexity. In all of the diagrams, the iSCSI traffic is clearly separated from the main network traffic.

4.3.1 Basic configuration

This simple Hyper-V network configuration includes iSCSI storage. It is in a simple environment with a few hosts that each has a few virtual machines. **This simple configuration is effective and achieves the same modular network configuration for both the management OS and guest virtual machines where the iSCSI data traffic is isolated from all other network traffic.**

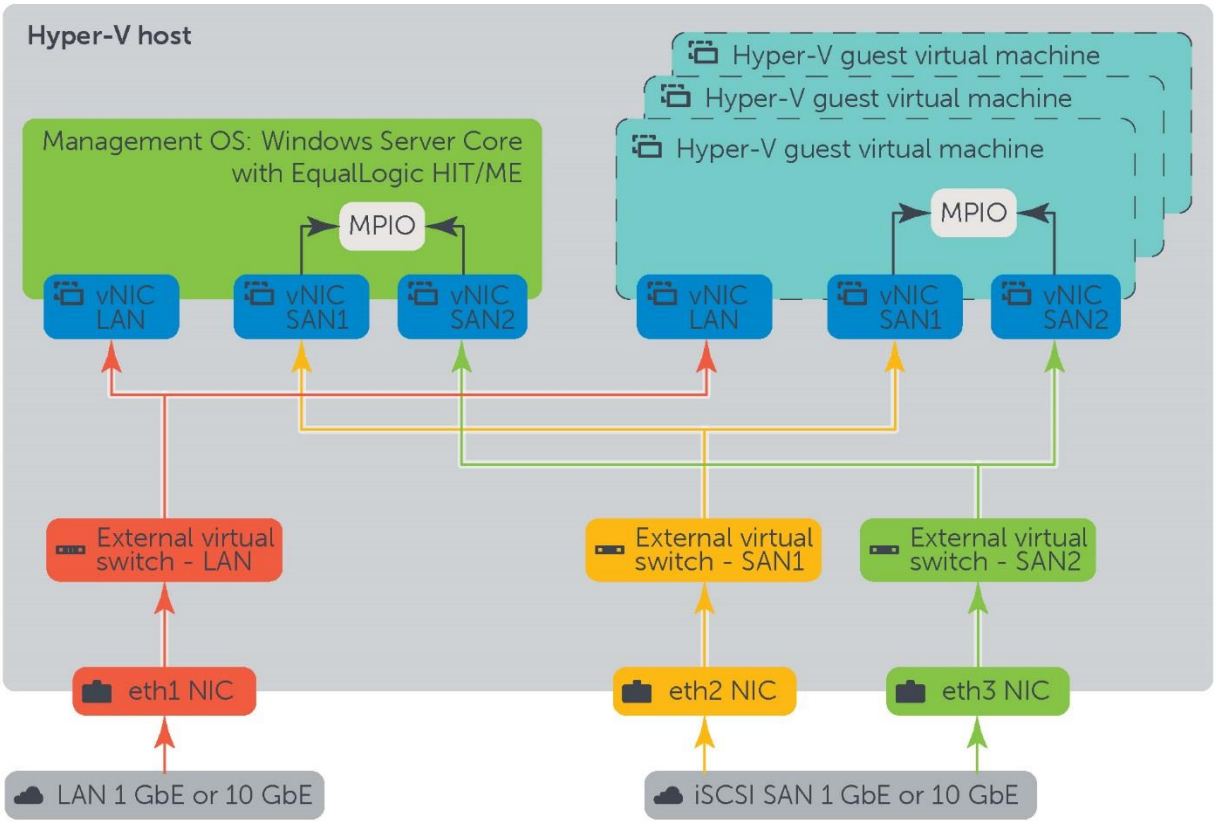

Figure 7 Simple Hyper-V network configuration

In Figure 7, the host has three physical NICs. The eth1 NIC is connected to the LAN network, and the other two NICs (eth2 and eth3) are connected to the iSCSI SAN Network. An external virtual switch has been created for each of the physical NICs on the host. For all of the virtual switches, the management OS has been allowed access because the management OS utilizes three virtual NICs (LAN, SAN1, SAN2) with each vNIC connected to a specific virtual switch. All network traffic in the management OS is converged through the LAN vNIC and virtual switch. The management OS is also attached to the iSCSI SAN through SAN1 and SAN2 vNICs and virtual switches to isolate the iSCSI traffic. In smaller environments the management OS will often house storage management and monitoring software that needs an actual storage connection to the iSCSI storage arrays on the iSCSI SAN. Because the iSCSI devices are presented down two paths (SAN1 and SAN2), the management OS will need to have multipath I/O (MPIO) installed for proper storage load balancing and failover.

Table 8 Possible management OS IP configuration scheme for the basic configuration

Physical	Virtual switch name	Management OS vNIC name	IP/subnet mask/gateway
eth1	LAN	vsLAN	10.127.4.50 / 255.255.252.0 / 10.127.4.1
eth2	SAN1	vsSAN1	10.10.6.210 / 255.255.0.0 / empty
eth3	SAN2	vsSAN2	10.10.6.211 / 255.255.0.0 / empty

In the above configuration, only the LAN vNIC is configured with a default gateway that can route to the external network. Niether SAN1 nor SAN2 vNICs have a default gateway, which means their traffic is not routable outside the 10.10.6.X network.

Another nice feature about this configuration is that the guest VMs use the same configuration with a single LAN vNIC and two vNICs (SAN1 and SAN2), . The common naming scheme provides easy identification and portability. As with the management OS configuration, the iSCSI traffic is isolated from the network traffic through the SAN1 and SAN2 vNICs. The guest VMs in this configuration have iSCSI storage provisioned directly to them; utilizing both SAN1 and SAN2 vNICs. Since dual paths are used, the guest VMs will also use MPIO.

Table 9 Possible Guest VM IP configuration scheme for the basic configuration

Physical	Virtual switch name	Guest vNIC name	IP/subnet mask/gateway
eth1	LAN	vsLAN	10.127.4.5x / 255.255.252.0 / 10.127.4.1
eth2	SAN1	vsSAN1	10.10.6.xx/ 255.255.0.0 / empty
eth3	SAN2	vsSAN2	10.10.6.xx / 255.255.0.0 / empty

Since this configuration is intended for hosts with only a few guest VMs, the same IP subnets can be used for the LAN and SAN vNICs. In the example above, four guest VMs could use a LAN IP of 10.127.4.51 through .54, a SAN1 IP of 10.10.6.xx through .yy , and a SAN2 IP of 10.10.6.xx through .yy. The simplicity of the configuration allows it to be easily scripted and copied from host to host.

This type of network configuration will work well in many SMB environments, especially if 10GbE networking is used for the LAN and iSCSI SAN. The key point to remember is that in this configuration, the iSCSI traffic is always physically isolated from all other network traffic for both the management OS and guest VMs by dedicating individual physical NICs and associated virtual switches and vNICs to the iSCSI traffic.

4.3.2 Converged network configuration

One of the issues encountered with the basic configuration is the inability to break out the cluster and live migration traffic on the Management OS. Hyper-V live migration traffic could result in congestion if it is not separated from other network traffic. A solution to this is to create one vNIC for the management traffic, one for live migration traffic and another for cluster heartbeat traffic. In this case, each management OS vNIC is connected to a port on the LAN virtual switch. Each management OS vNIC appears as a connection with its own protocol configurations such as IPv4 and IPv6 setting.

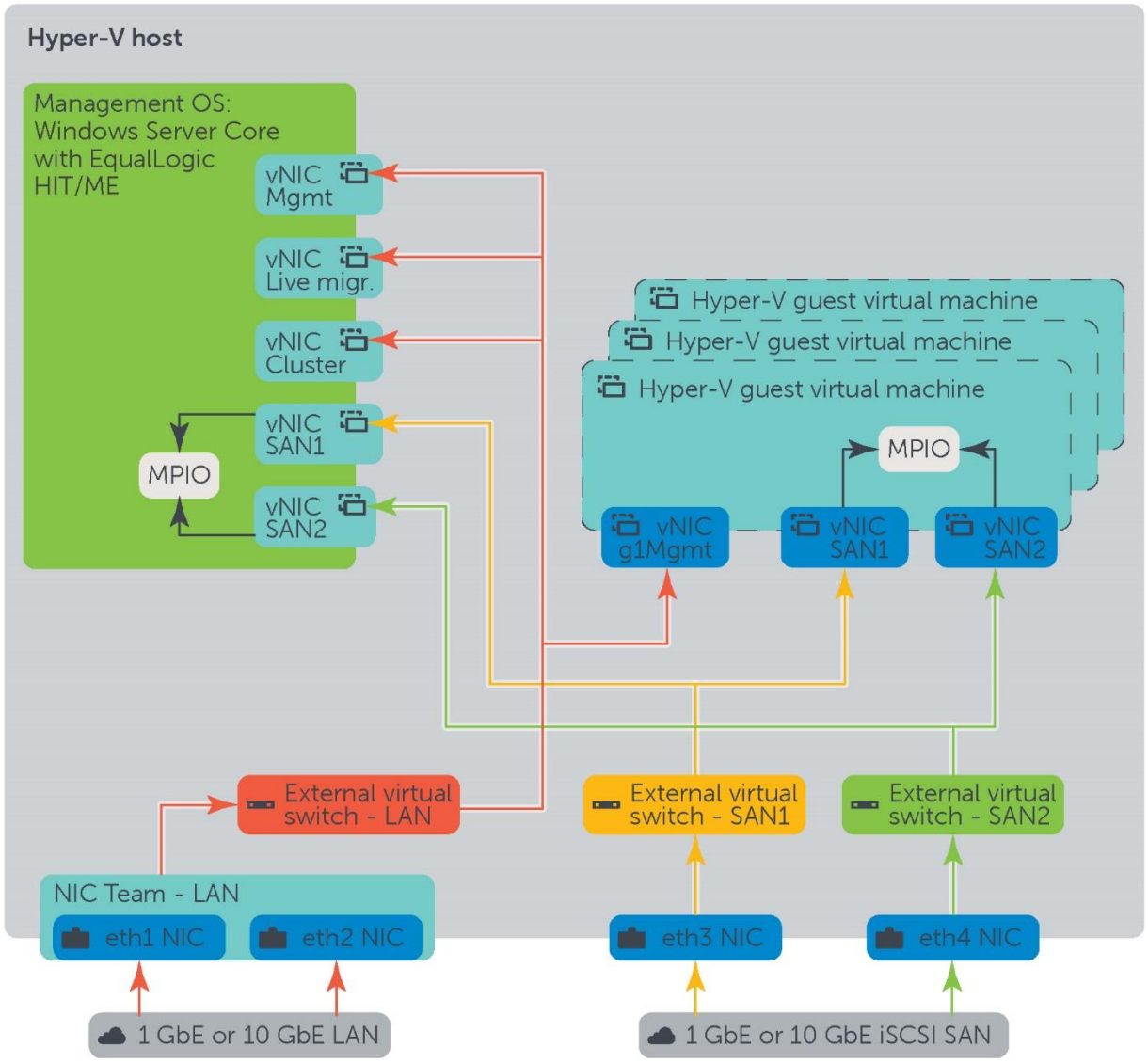

Figure 8 Converged network configuration

To further isolate these management OS vNICs on the physical network, assign VLAN IDs to them. However, this could add complexity to the configuration. Simplify by using QoS to assign each

management OS vNIC a weighted minimum bandwidth policy. Then, assign a default QoS policy to the virtual switch to reserve bandwidth for vNICs that do not have an explicit QoS policy.

Table 10 Possible management OS IP configuration

Physical	NIC team name	Virtual switch name	Mgmt OS vNIC name	QoS minimum bandwidth weight (example)	IP/subnet mask/gateway
eth1	LAN	LAN	Mgmt	5	10.127.4.50 / 255.255.252.0 / 10.127.4.1
eth2			LiveMigration	30	192.168.10.30/255.255.255.0/empty
			Cluster	20	172.126.5.1/255.255.0.0/empty
eth3	NA	SAN1	vsSAN1	NA	10.10.6.200 / 255.255.0.0 / empty
eth4	NA	SAN2	vsSAN2	NA	10.10.6.211 / 255.255.0.0 / empty

Note: There have been no changes from the basic configuration for the iSCSI portion of this configuration. The iSCSI traffic is still completely isolated from the LAN traffic by using iSCSI physical NICs, virtual switches, and vNICs. A possible guest VM IP configuration is shown in Table 11.

Table 11 Possible guest VM IP configuration

Physical	NIC team	Virtual switch name	Guest vNIC name	IP/subnet mask/gateway
eth1	LAN	LAN	g1Mgmt	10.127.4.5x / 255.255.252.0 / 10.127.4.1
eth2				
eth3	NA	SAN1	SAN1	10.10.6.x.x / 255.255.0.0 / empty
eth4	NA	SAN2	SAN2	10.10.6.xx / 255.255.0.0 / empty

In this configuration, note that the two physical NICs (eth1 and eth2) have been placed into a NIC team represented by the LAN virtual switch. This is usually done for LBFO and for bandwidth aggregation reasons (in this case, using the10GbE means the NIC team has up to 20 GbE of aggregated bandwidth). In addition, the iSCSI NICs have not been placed into a NIC team. It is important to recall that NICs used by iSCSI traffic cannot be used in a NIC team as iSCSI does not support NIC teaming. iSCSI requires MPIO for LBFO operations.

This configuration highlights the benefits of Hyper-V converged networking with respect to physical NIC reduction. Using a physical server would require seven NICs to achieve the same result as the configuration shown in Table 11 that uses four. It is estimated that each converged NIC saves 31 Watts. This translates into a power savings of 93 Watts for the server.

4.3.3 High performance iSCSI configuration

The high performance iSCSI configuration is essentially the same as the configuration in section 4.3.2. The difference is that the standard NICs used for the iSCSI SAN have been replaced with SR-IOV Enabled NICS.

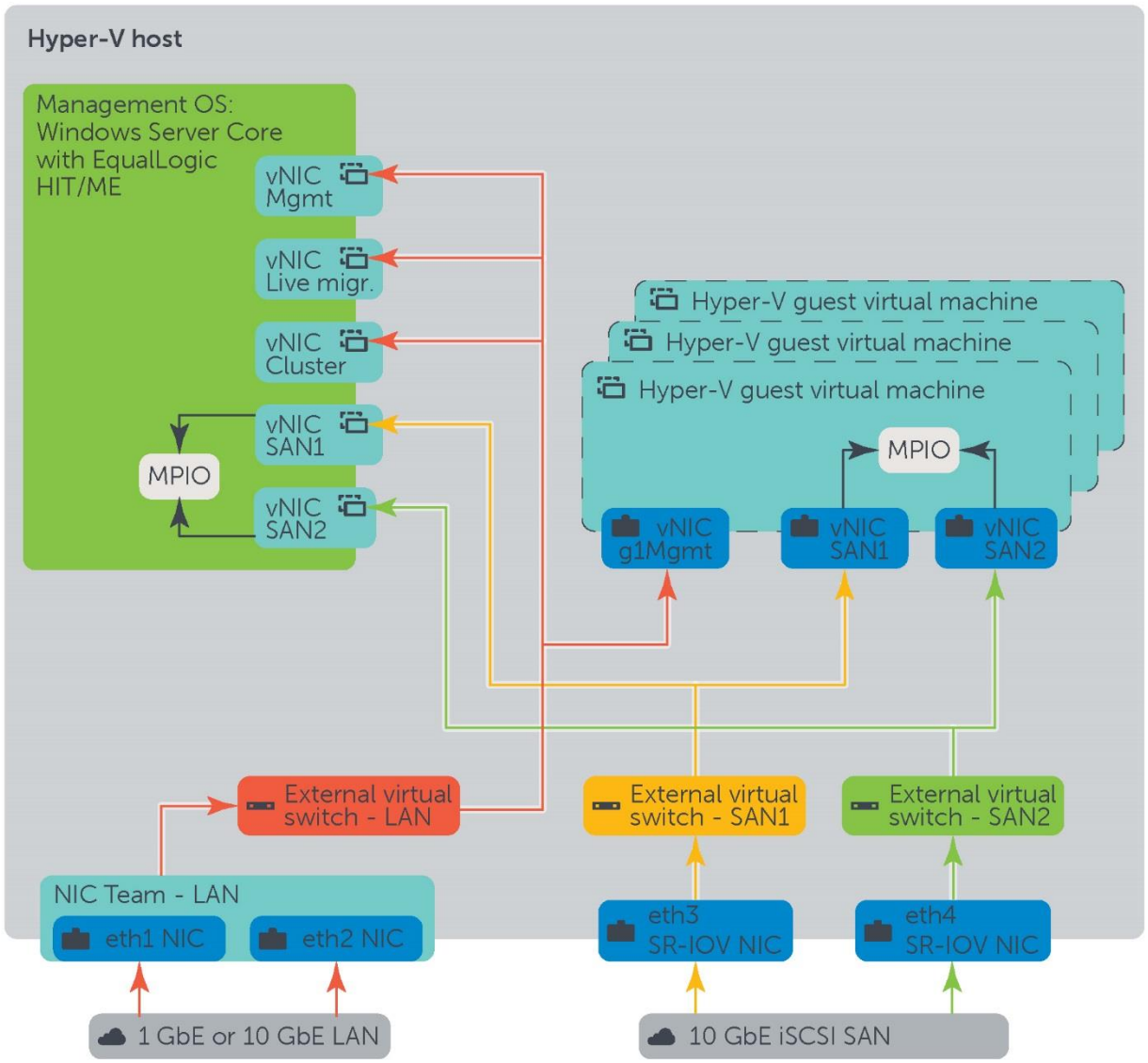

Figure 9 High performance iSCSI configuration

This configuration is used to increase the iSCSI performance to the guests, but has another advantage as well. In many environments, virtualized hosts become CPU bound before they become network bound. In theory, this configuration would free up significant CPU resources as the iSCSI data path processing would be handled directly by the SR-IOV NICs. More free CPU cycles make room for additional guest VMs.

5 Hyper-V networking summary

While each SMB environment has its own set of unique network and virtualization requirements, this paper has highlighted three designs that show various technologies and best practices. These designs provide iSCSI isolation, create multiple vNICs with separate QoS policies for the management OS, NIC teaming, and SR-IOV NIC implementation for high performance iSCSI traffic. Finally, the one rule applicable to Hyper-V networking for SMB is: **keep it simple**. Avoid introducing needless complexity into the environment when it is not needed.