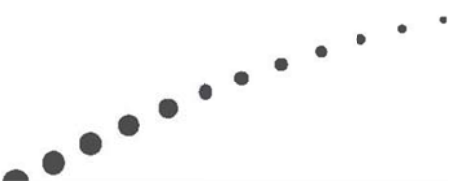




***Citrix XenServer Design:
Designing XenServer Network
Configurations***





Contents

- About..... 5
 - Audience..... 5
 - Purpose of the Guide 6
 - Finding Configuration Instructions..... 6
 - Visual Legend 7
 - Additional Terminology 8
- Chapter 1: Introduction 9
- Chapter 2: Basic XenServer Networking Concepts..... 11
 - Introduction to XenServer Networking 11
 - Connecting Virtual Machines to Networks..... 12
 - Networking Configuration after Installation..... 14
 - Impact of Pools on XenServer Networking 15
 - Sequence of Networking Configuration Tasks..... 18
 - Cabling Configuration for XenServer 18
 - Connecting XenServer to Physical Switches 21
- Chapter 3: Sample Networking Scenario 22
 - Example: Adding Virtual Machines to a Network..... 22
 - Creating Network Resiliency through Bonds..... 23
 - Connecting a VM to a Network using Virtual Interfaces..... 25
 - Segregating VM Traffic from Management and Storage Traffic..... 27
 - Scenario 1: Segregating Traffic..... 28
 - Scenario 2: Using the Management Network for VM Traffic..... 29
 - Scenario 3: Isolating VM Traffic on a Private Network..... 30
 - Scenario 4: Connecting VMs to Multiple Linked VLANs 34



- Chapter 4: Specifying Networking Requirements..... 40
 - Overview 40
 - Introduction..... 41
 - XenServer Networking Support and Requirements 42
 - Defining Your Networking Requirements 43
 - Considering Workload Communication Requirements..... 43
 - Evaluating Your Current Network Configuration 45
 - Determining Host Networking Requirements..... 46
 - Reviewing Initial Pool Design against Networking Requirements 47
 - Calculating the Number of Physical NICs per Host 49
 - Calculating Bandwidth Requirements 50
- Chapter 5: Designing XenServer Networks..... 52
 - Overview 52
 - Deciding to Use the Distributed Virtual Switch..... 53
 - Designing Network Redundancy..... 56
 - Considering NIC Bonding..... 56
 - Selecting a Type of NIC Bonding..... 58
 - Understanding Active-Active NIC Bonding..... 58
 - Understanding Active-Passive NIC Bonding..... 60
 - Bonding Management Interfaces and MAC Addressing..... 62
 - Best Practices for Bonded Interfaces 62
 - Designing Networks for Performance..... 63
 - Testing XenServer Network Performance 65
 - Limiting Bandwidth Consumption for High Demand Workloads..... 66
 - Additional Considerations 68
 - Enabling Promiscuous Mode for Traffic Monitoring..... 68



Chapter 6: Designing Your Storage Network Configuration.....	70
Overview	70
Creating a Separate Storage Network.....	71
Assigning IP Addresses to NICs (Management Interfaces)	73
Configuring Redundancy for Storage Traffic	75
Choosing to Enable Multipathing Support or Bond NICs.....	75
Suggestions for Improving Storage Network Performance	77
iSCSI Storage.....	77
Configuring Networks with Jumbo Frames	78
Chapter 7: Considering Network Performance for PVS – XenServer Deployments	81
Virtualizing the Provisioning Services Server	81
IP Addressing Requirements.....	83
Isolating the Provisioning Services Streaming Service	83
Disabling the Spanning Tree Protocol.....	83
Best Practice Configurations	83
Chapter 8: Verifying Your XenServer Networking Configuration	85
Verifying XenServer Networking on a Pool	85
Verifying your Physical Configuration	85
Verifying your XenServer Networking Settings and Configuration	87
Resolving Issues	88
Revision History	90

About

This guide helps you understand design your XenServer networking and design a networking configuration for XenServer environments. It includes the following topics:

- Best practice information about the primary management interface, NIC bonding, jumbo frames, and storage networks
- High-level information about features you may want to enable as part of your networking configuration, such as the Distributed Virtual Switch solution
- The correct sequence in which to configure XenServer networking, including guidance about cabling XenServer hosts and connecting them to physical switches
- Checklists to help you gather requirements for your XenServer networking configuration

Audience

Before reading this guide, you should have a basic knowledge of networking. This guide has several audiences:

- **Systems Architects.** Systems architects who are designing a virtualized environment.
- **Infrastructure Engineers and Network Administrators.** Networking and storage professionals who configure storage or manage the Layer 2 network infrastructure in their organizations.
- **Application Administrators.** XenApp and XenDesktop administrators who are implementing a virtualization solution to virtualize Citrix products, IT infrastructure, or other applications they manage.



This guide assumes that you are familiar with basic XenServer concepts, including XenServer installation, XenCenter, resource pools, and the pool master.

Purpose of the Guide

This guide is meant to provide you with the best-practice information you need to design your XenServer networks.

To provide you with the foundation you need to understand the recommendations, the first half of the guide provides an explanation of XenServer networking concepts using a scenario-based approach.

The second half of the guide provides you with information to help you select between various XenServer networking options and information about the best ways to configure them.

Because this is a design guide, it generally does not provide configuration instructions except as needed to clarify concepts. As the most common way of managing XenServer and XenServer pools is through XenCenter, this guide mainly refers to XenCenter and XenCenter help, unless specified differently.






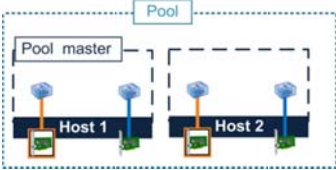
Finding Configuration Instructions


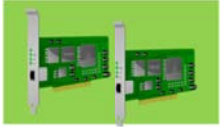
You can find networking configuration instructions in the following locations:

- **XenCenter Help.** The XenCenter help provides UI-based step-by-step instructions using XenCenter, the XenServer UI-based administration console. Users who are not comfortable with the XenServer `xe` commands, may prefer this option.
- **XenServer Administrator's Guide.** The *XenServer Administrator's Guide* provides command-line based instructions for performing networking tasks. For integrators, it also provides information about XenServer networking from the object-model perspective.

Visual Legend

This guide relies heavily on diagrams to explain key concepts. These diagrams use the following icons:

Icon	Meaning
	<p>Virtual Machine. A virtual computer that runs on the XenServer host.</p>
	<p>Virtual Interface. On a VM, there is a logical interface that appears and functions like a NIC; this interface is known as a <i>virtual interface</i>. A virtual interface lets VMs send and receive network traffic. Some product literature refers to virtual interfaces as <i>VIFs</i> and <i>virtual NICs</i>.</p>
	<p>Network. A network is the virtual network switching fabric built into XenServer that lets you network your virtual machines. It links the physical NICs to the virtual interfaces and connects the virtual interfaces together.</p>
	<p>Host. A XenServer host is the physical server on which the XenServer hypervisor is running.</p>
	<p>NIC. The physical network card (NIC) in your host.</p>
	<p>Pool. A XenServer resource pool or “pool” is a connected group of hosts which provides a platform on which virtual machines run.</p> <p>To join hosts to a pool, they require broadly compatible hardware and must be running the same XenServer version and patches.</p> <p>Pools comprise a pool master and subordinate servers known as pool members (sometimes also referred to as "slaves"). The pool master</p>

	<p>provides a single point of contact for all the servers in the pool and the master will forward commands to individual pool members as necessary.</p>
	<p>Physical Switch. The device on a physical network that connects network segments together.</p> <p>This guide may present physical switches either as a three-dimensional physical box or as a one-dimensional panel with ports.</p>
	<p>NIC Bond. In this guide, enclosing NICs in green represents a bond.</p> <p>A NIC bond is a pair of NICs configured so they logically function as one network card. NIC bonding is also known as <i>NIC teaming</i>.</p>

Additional Terminology

These terms appear in the sections that follow:

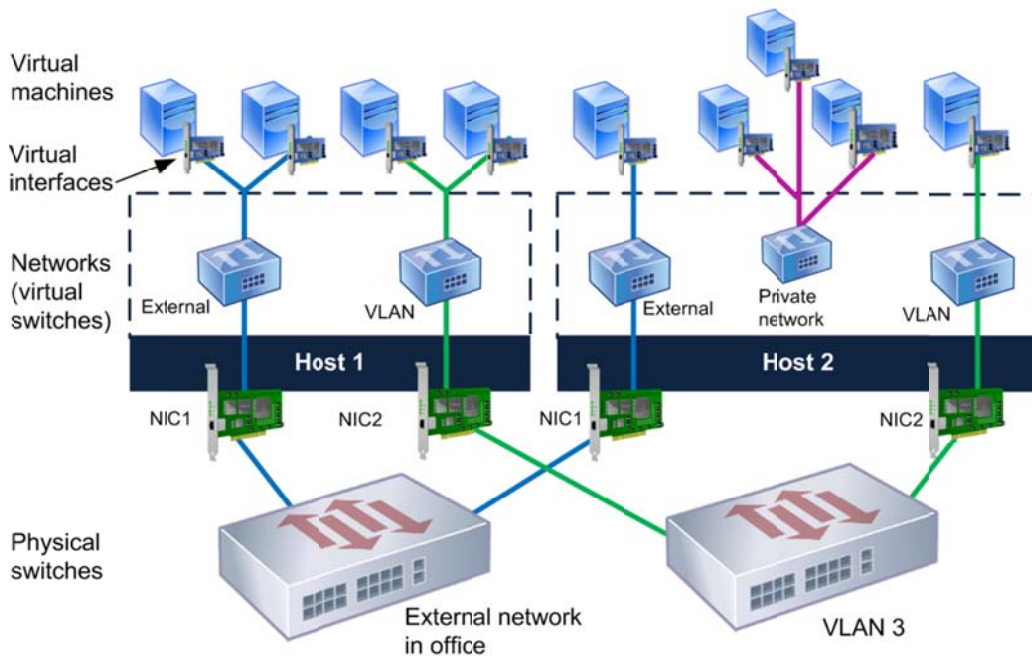
Primary Management Interface. The primary management interface is a NIC assigned an IP address that XenServer uses for its management network, including, but not limited to, traffic between hosts, between a host and Workload Balancing and for live migration.

VM traffic. VM traffic refers to network traffic that originates or terminates from a virtual machine. This is sometimes referred to as guest traffic or VM/guest traffic.

Chapter 1: Introduction

This documentation explains basic networking concepts and their application by using a series of scenarios to illustrate the concepts. The scenarios begin immediately after installation and end with connecting a VM to a network.

These sample scenarios focus on three different types of networks: External Networks, VLANs, and single-server private networks. If you configured the scenarios demonstrated in this guide, by the time you finished, you would create a deployment that looked like the following illustration.



This illustration shows how virtual machines connect to three different types of networks: an external network, a VLAN network, and a single-server private network.



This guide explains these types of networks by providing the following information:

Chapter 2 introduces XenServer networking and explains how to prepare for XenServer networking configuration by configuring the physical infrastructure and hardware layers in your environment, including the correct sequence for physically configuring networking. The chapter also discusses the effect pooling XenServer hosts has on networking and describes the networking configuration after installation.

If you want to read a list of XenServer networking definitions before reading this information, see the Visual Legend on page 7. Otherwise, the definitions are provided in chapter 2 as you read the sections.

Chapter 3 provides several sample scenarios that illustrate how to add virtual machines to a network. The first scenario guides you through the process of segregating different types of traffic, including storage and management traffic. The second scenario gives you an alternative to dedicating NICs to specific types of traffic; it shows an example of using the management network for management and VM traffic. The third scenario shows an example of how to segregate traffic by creating a single-server private network on a host.

Chapter 4 provides guidance about how to determine what networking configurations and hardware your XenServer deployment will require. The chapter includes tables with references to relevant information to help you jump to key information about how to define your NIC configuration, calculate bandwidth requirements, and other topics.

Chapter 5 provides information about key design choices you will make when creating your XenServer networks, including whether to use the Distributed Virtual Switch, how to configure XenServer network redundancy (NIC bonding), and how to design XenServer networks for performance.

Chapter 6 provides information about how to create a separate physical network for storage traffic, set an IP address on a NIC, configure iSCSI multipathing, and configure support for jumbo frames. This chapter also includes suggestions for improving storage network performance and best practices.

Chapter 7 provides information about how to virtualize a Provisioning Services server on XenServer using SR-IOV. This chapter also provides some references to best practices for XenServer-Provisioning Services deployments.

Chapter 8 provides information about how to verify your XenServer networking configuration after you physically configured it. This chapter provides a process for verifying networking on a host and on pools as well as information about resolving issues.

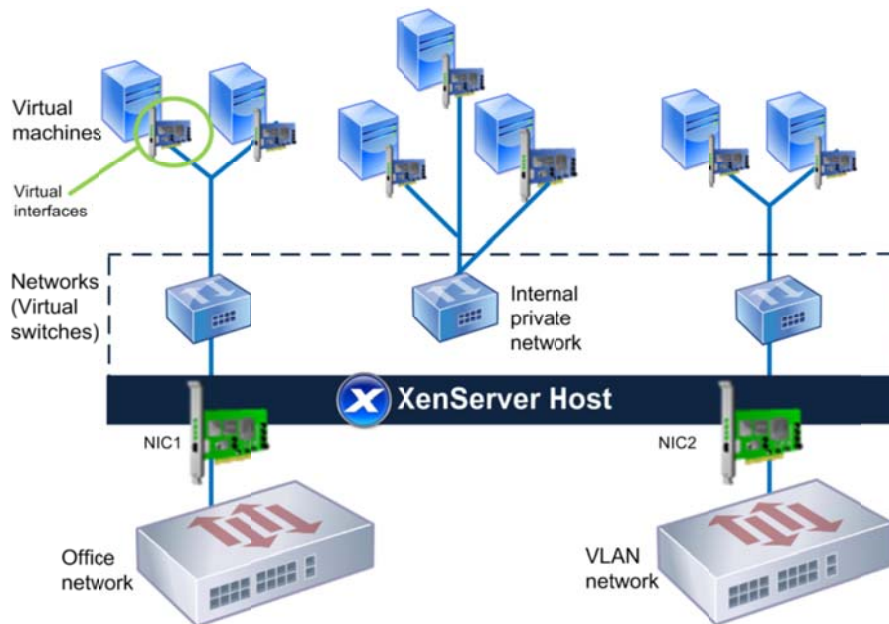
Chapter 2: Basic XenServer Networking Concepts

This chapter includes the following topics:

- An introduction to XenServer networking
- The network settings created during installation

Introduction to XenServer Networking

XenServer provides virtual networking features that let you build networks for your virtual machines the same way you build networks for physical machines.



The VMs connect to three different types of networks: an office network, an internal private network, and a VLAN.

You can connect virtual machines to your production network like you connect physical machines or build private networks within a host or pool for testing, development, or security purposes. You can connect virtual machines to your VLAN networks using standard VLAN configurations.

The most important networking components XenServer lets you configure are *virtual interfaces* and *networks*:

- **Virtual interfaces.** Virtual machines connect to networks using virtual NICs, known as virtual interfaces. Virtual interfaces let VMs send and receive network traffic. You can assign each virtual interface its own IP address and MAC address. Some product literature refers to virtual interfaces as *VIFs* and *virtual NICs*.
- **Networks.** XenServer has an internal virtual switch, known as a network, that lets virtual machines on a XenServer host communicate with each other using the same networking protocols that are used on physical networks.

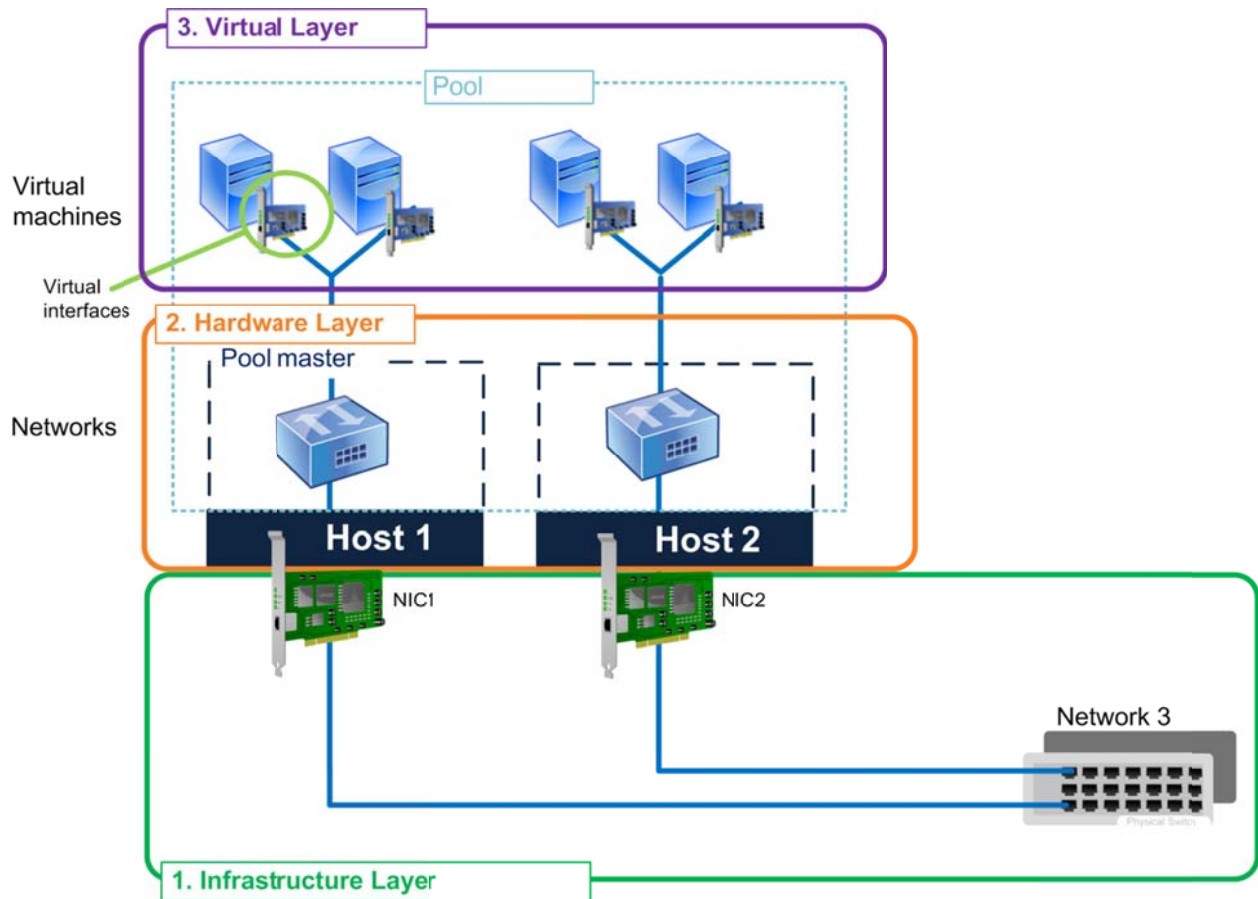
A network is the logical network switching fabric built into XenServer that lets you network your virtual machines. It links the physical NICs to the virtual interfaces and connects the virtual interfaces together. These networks are virtual switches that behave as regular L2 learning switches. Some vendors' virtualization products refer to networks as *virtual switches* or *bridges*.

Connecting Virtual Machines to Networks

When you are configuring network connectivity on XenServer hosts, your ultimate goal is to connect the VMs to a network. To do this:

1. Connect the host to a physical network. (For VMs without external network connectivity, you would configure a private network instead.)
2. Connect the VM by creating a Virtual Interface for it and connecting the Virtual Interface to a network. As shown in the illustration on page 11, the virtual interfaces on the VMs connect to networks in a host and then connect to a physical network through the host's NIC.

One way to think about these tasks is that you need to configure connectivity at both the hardware and virtual layers as shown in the illustration that follows.



This illustration shows the order in which you should configure networking in your virtual environment: (1) Start on the physical infrastructure layer, which means connecting NICs to switches; (2) configure the hardware layer, which means connecting hosts to networks and configuring these networks; (3) configure the virtual layer, which means attaching VMs to networks through virtual interfaces.

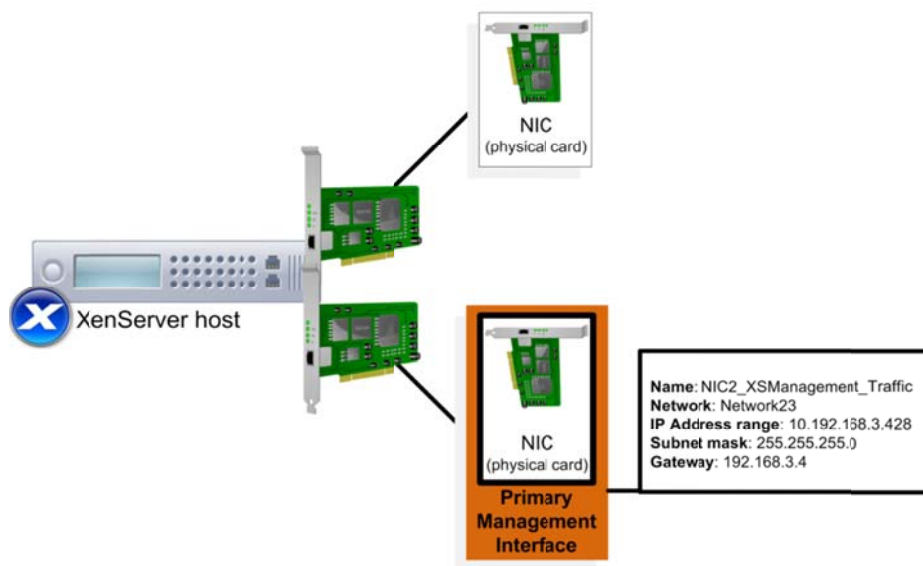
Important: Configuring networking in the order listed described in “Sequence of Networking Configuration Tasks” on page 18 is critical. If you vary from this sequence, the primary management interface may not be configured correctly on each host. If this occurs, all VMs in the pool may start on the pool master and not their home or optimal servers.

Networking Configuration after Installation

After installation, the XenServer *host* has all the information it needs to connect to at least one of your external networks. This is because you define the following networking options while installing XenServer:

- IP Address Configuration and Other Settings.** You set the host's initial XenServer networking configuration when you first install XenServer on the physical computer. XenServer Setup configures options, such as the IP address configuration (DHCP/static), based on the values you provide during installation.
- Network Connectivity.** XenServer installation prepares each NIC connected to a switch for network connectivity by creating one network for each NIC. This means that if the host has, for example, three NICs, XenServer creates three networks: Network 0, Network 1, Network 2. For a visual explanation, see page 15.
- Primary Management Interface and the Management Network.** During XenServer Setup, you specify an IP address for one NIC. XenServer uses that NIC to connect to your organization's network and to carry management traffic for functions like communicating with other hosts in a pool, XenCenter, Workload Balancing, and other components. This NIC is known as the *primary management interface*. This is the only NIC that Setup configures with an IP address.

The illustration that follows shows a regular (unconfigured) NIC and a NIC configured as a primary management interface.

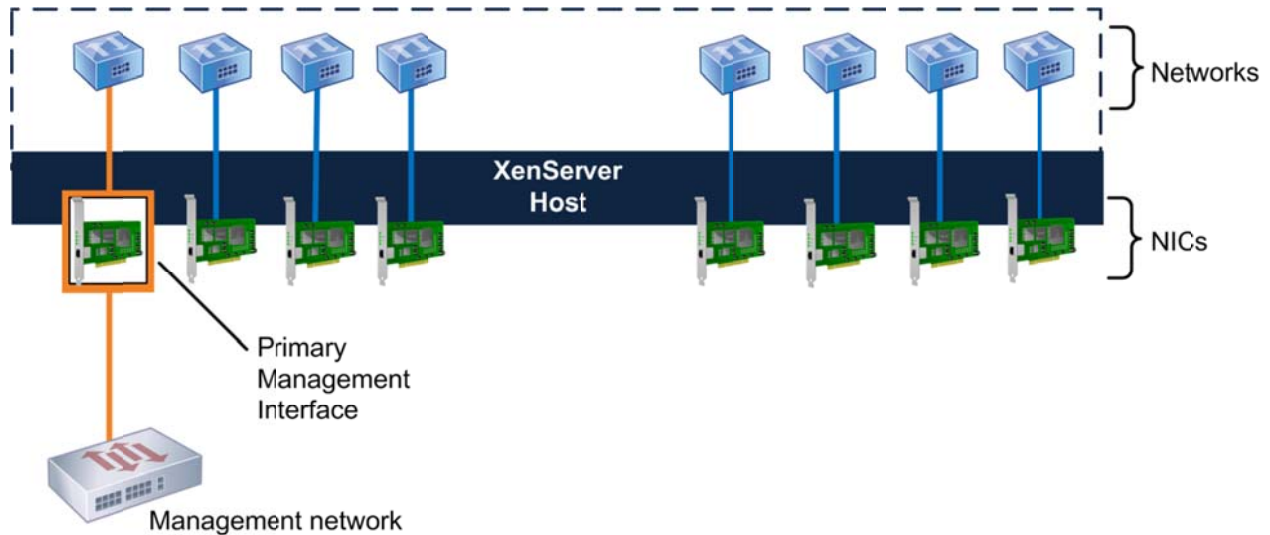


This illustration contrasts a regular NIC with one configured as the primary management interface. The primary management interface has an IP address, subnet mask, and gateway assigned to it.



During installation, XenServer also creates a separate network for each NIC it detects on the host. Unless you change this set up, XenServer uses the additional NICs on the host for VM traffic only.

The illustration that follows shows an example of XenServer's initial network configuration following installation.



This illustration shows how, during installation, XenServer lets you choose a NIC as the primary management interface. In this case, the administrator selected NIC0. XenServer uses the other NICs for VM traffic.

Most environments require additional configurations to these basic network settings. These can range from creating pools to integrating additional networks, connecting your VMs to those networks, and configuring a separate storage network. The scenarios in the following chapter provide examples of these tasks.

Note: If you plug any NICs into switches after installing XenServer, if you cannot see the NICs in XenCenter or `xconsole`, you might need to either a) run `xe pif-list` or `xe pif-plug` in the CLI or reboot the XenServer host.

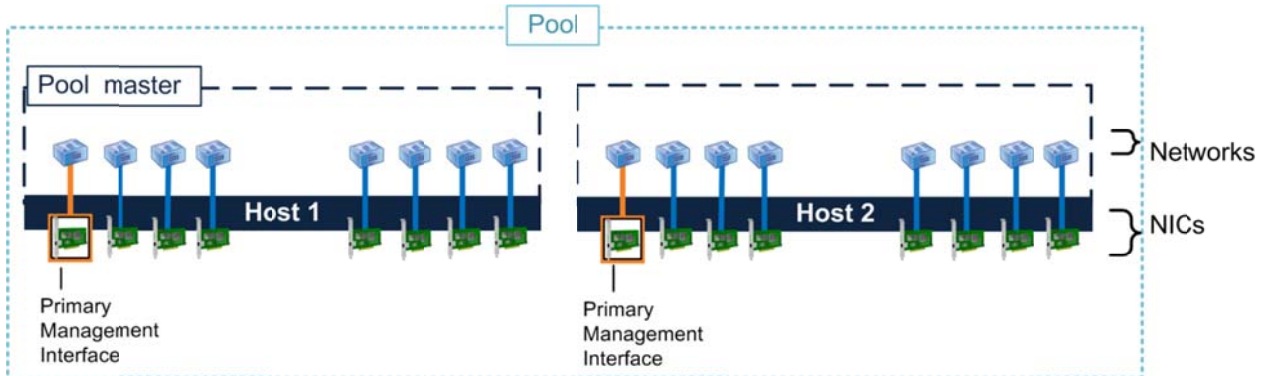
Impact of Pools on XenServer Networking

Networking is a pool-level feature in XenServer. When you change networking on the pool master, XenServer synchronizes all hosts in a pool to use the same network settings.

As a result, for XenServer to operate correctly, you must ensure that network settings match across all hosts in the pool, including:

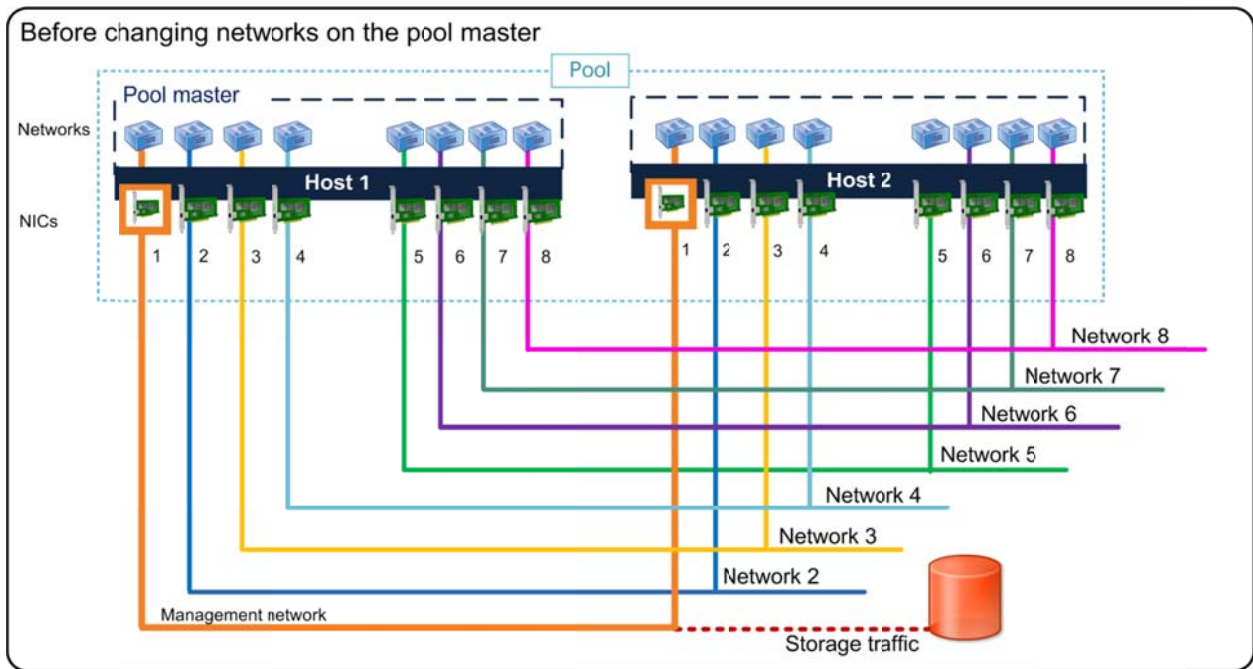
- Which NICs are bonded
- Which NICs are configured as the primary management interface
- Which NICs connect to storage

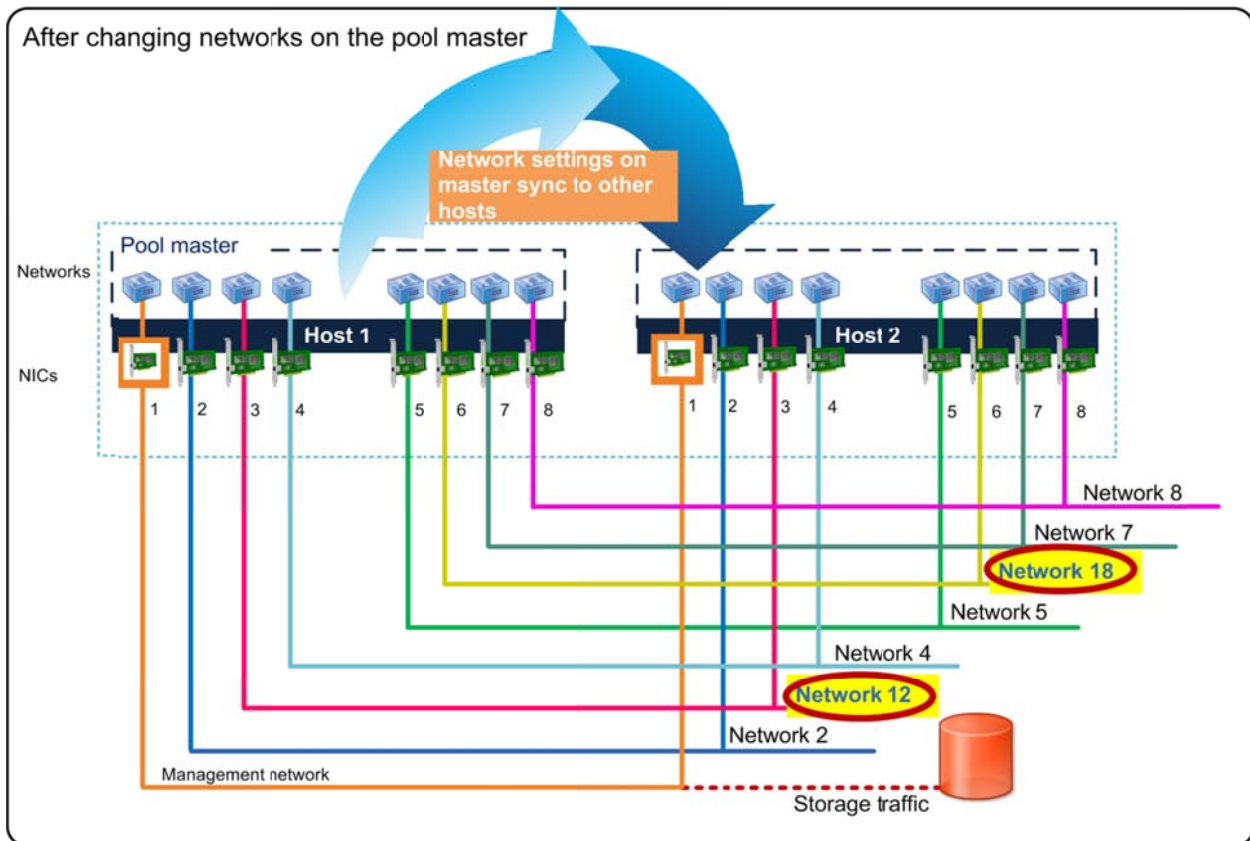
The networks to which NICs connect must be the same on the corresponding NICs on each host in the pool.



This illustration shows two hosts joined together in a pool before any networking configuration is performed on them.

Ideally, you should add all desired hosts to the pool before configuring any network settings. Pooling the hosts before configuring networking creates cleaner records in XenServer's internal networking-configuration database.





These two illustrations show how XenServer replicates the network settings created on the pool master on all other hosts in the pool. In the top illustration, NICs 3 and 6 on both hosts use Networks 3 and 6. In the bottom illustration, after reconfiguring NIC 3 on the pool master to use Network 12 and NIC 6 to use Network 18, XenCenter automatically configures the other host in the pool to use those settings.

After creating a new pool or joining a host to an existing pool, XenServer automatically replicates the network settings on the master to the joining hosts.

When you use XenCenter to make networking changes, XenCenter changes the other hosts to match the newly modified host. When you use the CLI to change network settings, you must either:

- Change each host manually to match the modified host's settings
- Make the change on the pool master and restart all the member hosts in the pool

XenServer requires network settings to match across the pool because of features that use live migration, such as XenMotion, High Availability, and Workload Balancing. These features enable the physical server hosting a VM to change at any time, and possibly automatically without your intervention. Therefore, the VMs must be able to access all of their target networks regardless of which host XenServer moves them on to.



For this reason, it is critical to have *and maintain* an identical physical cabling, NIC, and switch configuration for each host across the pool. Likewise, Citrix strongly recommends changing the physical configuration on all hosts in a pool before changing network settings on the pool.

Important: After joining the hosts to the pool, check the primary management interface on each member host to make sure that it has its own unique IP address and/or set the correct static IP address.

Sequence of Networking Configuration Tasks

Citrix recommends performing your initial networking configuration in the sequence that follows to help ensure XenServer stores your networking configuration correctly:

1. Cable the hosts by plugging all NICs into the appropriate switches, as described in “Cabling Configuration for XenServer” on page 18.
2. Configure the switches. See “Connecting XenServer to Physical Switches” on page 21.
3. Install XenServer on the hosts. Citrix recommends that you ensure your networking configuration is set up correctly before creating a resource pool, since it is usually easier to recover from a bad configuration in a non-pooled state. To verify networking is set up correctly, see “Chapter 8: Verifying Your XenServer Networking Configuration.”
4. Create a pool of the hosts, if you want to pool them. See “Impact of Pools on XenServer Networking” on page 15.
5. Configure NIC bonds and networks. For more information, see the scenarios in “Chapter 3: Sample Networking Scenario.”

Important: Do not configure the XenServer High Availability feature until after you complete your networking configurations. Networking configuration can interrupt the High Availability heartbeat and cause hosts to shut themselves down. Consequently, the hosts probably will not reboot correctly and will need the **host-emergencyha-disable** command to recover.

Cabling Configuration for XenServer

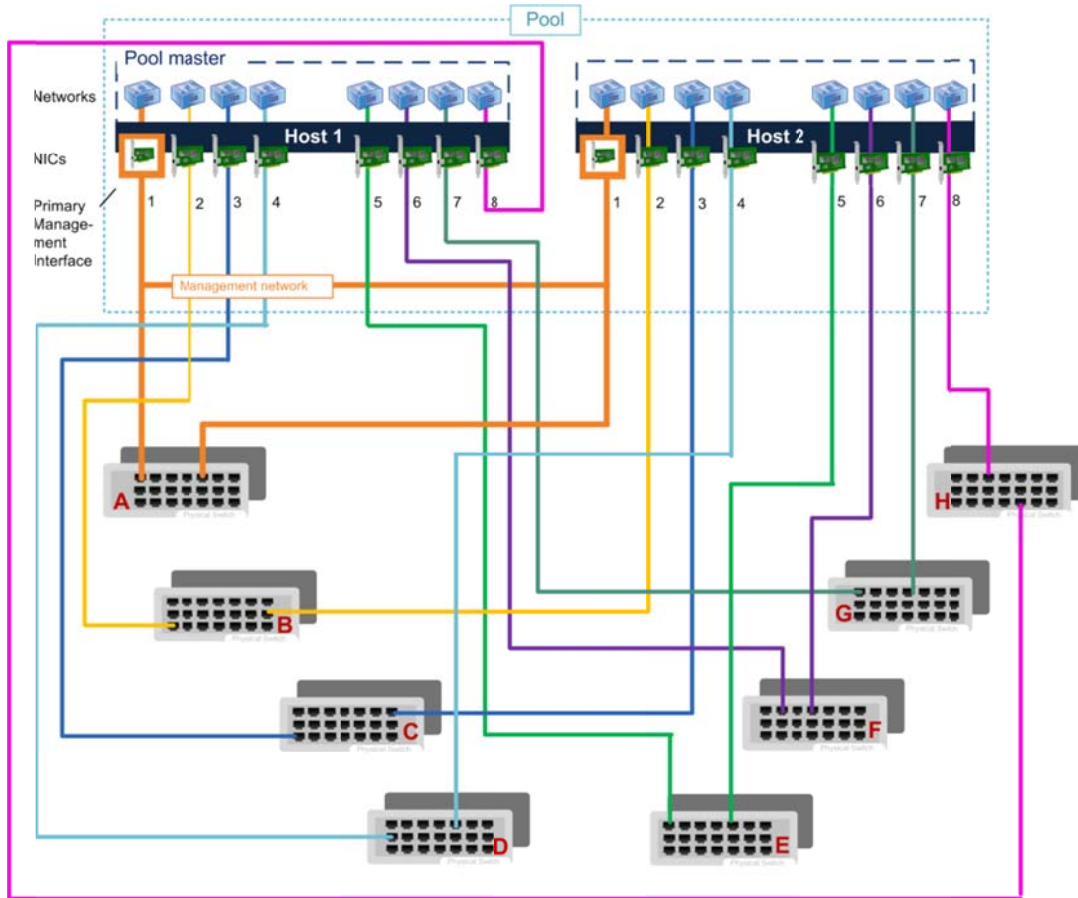
Citrix recommends plugging the physical Ethernet cables into all the NICs and the appropriate switches **before** installing XenServer. The ideal process is as follows:

1. If you did not cable your hosts before installation, plug all the NICs in each host in the pool into the appropriate switch ports.
2. Connect the corresponding NICs on each host in the pool to the same physical switch (that is, the same subnet).

The term *corresponding* refers to the NIC of the same number on another host. For example, NIC 3 on Host 1, NIC 3 on Host 2, NIC 3 on Host 3. This means that each individual NIC

on every host must connect to the same physical network as the NIC in the same position on all other hosts in the pool.

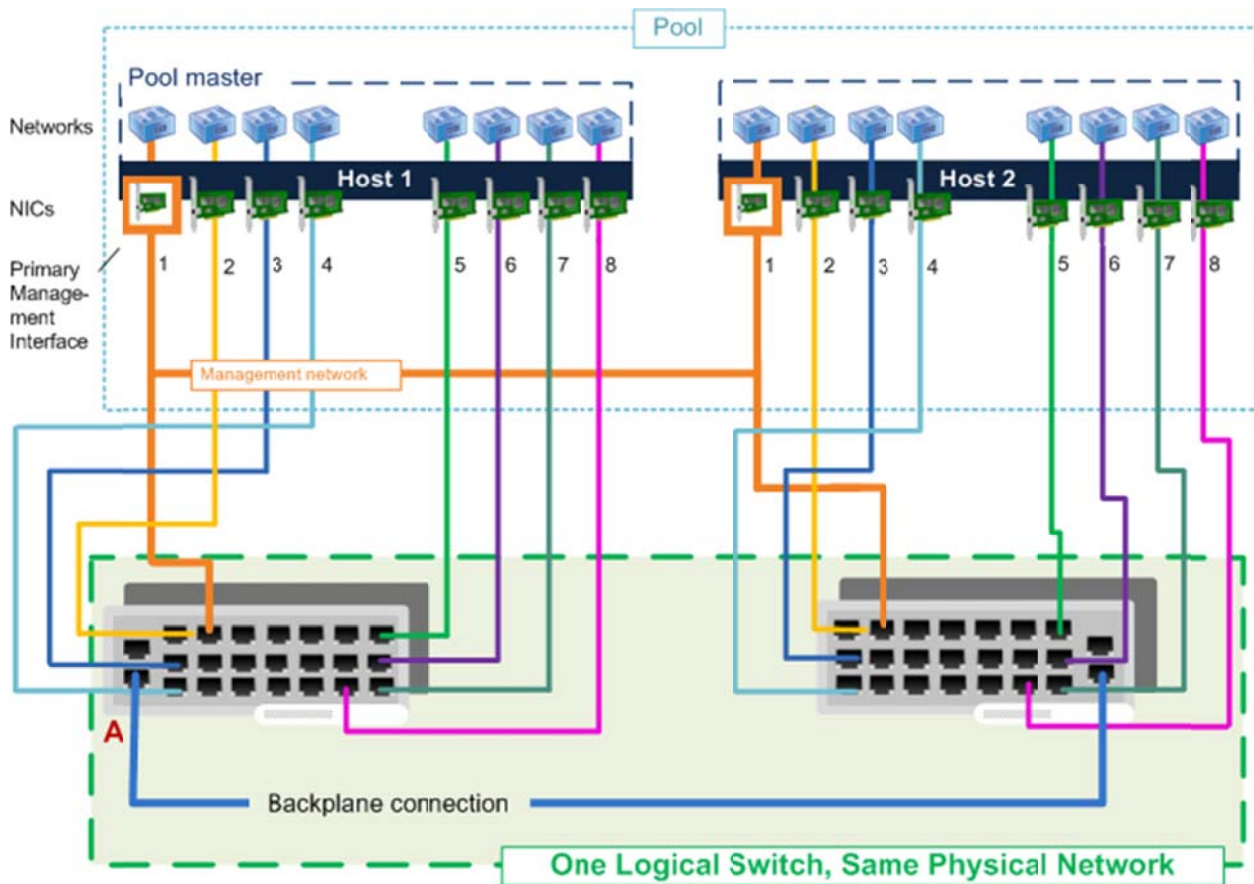
The following figure is a visual example of this configuration in an enterprise environment.



This illustration shows how each corresponding NIC on both hosts must physically connect to the same network. Each switch represents a separate physical network. Each member host's NICs must be connected to the same physical networks as the corresponding NICs on the pool master.

Ensuring the cabling on each host in the pool is correct is critical. As shown in the previous illustration, all NICs must connect to the same physical networks (shown as separate switches) as the NICs in the same position on all hosts across the pool.

In an environment with only one logical switch (for example, one that has a hierarchy of switches that form one large physical network), you only need to connect the NICs to switches on that network that have the same physical or logical (VLAN) connectivity. The example that follows shows how you might cable such an environment.



This illustration shows two switches that are connected across a backplane and are on the same physical network. These switches function logically as one unit. Because there are no VLANs configured on any of the ports and all ports have the same connectivity, the NICs can be plugged into any port on these two switches.

XenServer cannot detect if you make any errors while setting up the physical network. For example, if a XenServer host expects to be able to contact a specific gateway using a certain NIC, XenServer cannot indicate the cabling is incorrect. If you receive errors, they might not indicate network configuration as the cause.

Ensuring that the corresponding NIC on each host has the same network configuration is what ensures that a host's VM attached to, for example, Network 1, can communicate with a VM attached to Network 1 on another host. This ensures that if you migrate a VM to a new host, the VM retains the same physical connectivity after migration.

Note: When you configure networking, if you do not have all of your NICs plugged in to switches, you must have, at a minimum, the NIC(s) for the primary management interface on all hosts in your pool plugged into your network. Otherwise, the pool master cannot synchronize its network settings to the member hosts. Likewise, if you are using a dedicated NIC for storage, you must also connect the cables for that NIC on each host.



Connecting XenServer to Physical Switches

When you are connecting a XenServer host to a switch, configure the switch's ports differently than you would when connecting a workstation to a switch. There are specific, critical guidelines about the Spanning Tree Protocol (STP) and enabling PortFast.

For more information, see CTX123158 -- [*Considerations for XenServer Switch Ports*](#).

Chapter 3: Sample Networking Scenario

This chapter provides a scenario-based example of how to connect virtual machines to a physical network. This includes the following:

- Segregating traffic
- Using the management network for traffic in a very small environment

Example: Adding Virtual Machines to a Network

This section provides a sample scenario of a simple networking configuration that includes connecting VMs to networks, creating redundancy, and configuring NICs.

Designing a XenServer networking deployment may require several tasks, including, for example, configuring redundancy for network availability, configuring NICs, and, ultimately, connecting VMs to the desired networks. During this process, you might also separate different types of traffic for security or performance reasons (for example, separating traffic for managing the XenServer platform from VM traffic).

Before configuring networking on a pool, you should know to which networks your VMs will need to connect. A standard network configuration process might require:

1. Configuring redundancy for network availability.
2. Creating separate storage or management networks (used to separate management or storage traffic from VM traffic).
3. Creating VMs and connecting them to the desired XenServer network(s).

This section provides you with an example of that process. This section describes the different configuration options and steps required to put your virtual machines on the network by using a

sample scenario. While the scenario might not directly apply to your environment, it is designed to put XenServer’s networking features into context.

Creating Network Resiliency through Bonds

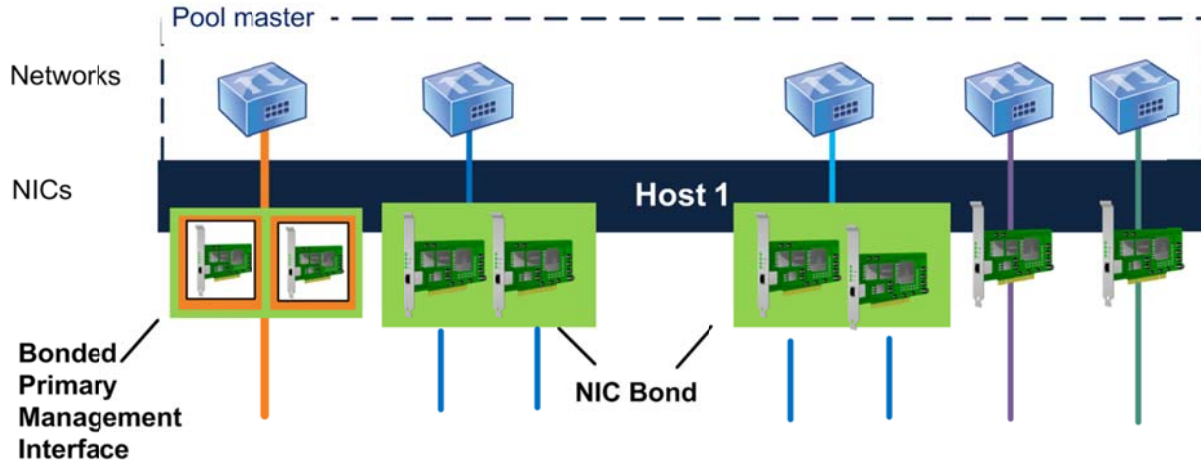
After joining all hosts to your pool, you may want to ensure that any critical servers have high availability access to the network. One way XenServer lets you achieve high network availability is to create redundancy through *NIC bonding*.

NIC bonding is a technique for increasing resiliency and/or bandwidth in which an administrator configures two NICs together so they logically function as one network card. Both NICs have the same MAC address and, in the case of management interfaces, have one IP address.

XenServer supports *bonding* two NICs together on a host. If one NIC in the bond fails, XenServer automatically redirects traffic to the second NIC. NIC bonding is also sometimes known as *NIC teaming*.

You can use XenCenter or the xe CLI to create NIC bonds. If XenCenter is managing a pool, XenServer automatically replicates the bonding configuration across all hosts in the pool.

In the illustration that follows, the primary management interface is bonded with a NIC so that it forms a bonded pair of NICs. XenServer will use this bond for management traffic.



This illustration shows three pairs of bonded NICs, including the primary management interface. Excluding the Primary Management Interface bond, XenServer uses the other two NIC bonds and the two un-bonded NICs for VM traffic.

Ensuring Resilience through Redundant Switches

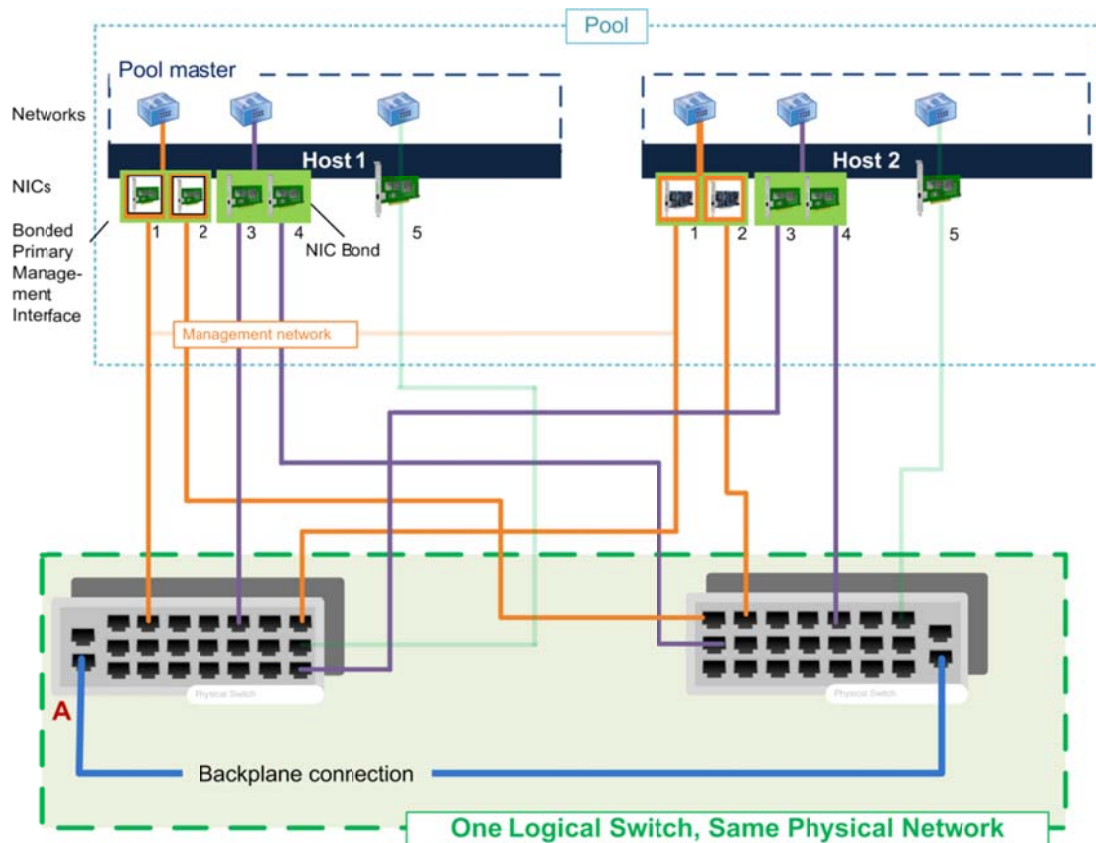
When VM networks use bonded NICs, traffic is sent over both NICs. If you connect one of the NICs in a bond to a second (redundant switch) and a single NIC or switch fails, the virtual machines remain on the network since their traffic fails over to the other NIC/switch.

Provided you enable bonding on NICs carrying only guest traffic, both links are active and NIC bonding can balance each VM's traffic between NICs. Likewise, bonding the primary management interface NIC to a second NIC also provides resilience. However, only one link (NIC) in the bond is active and the other remains unused unless traffic fails over to it.

If you bond a management interface, a single IP address is assigned to the bond. That is, each NIC does not have its own IP address; XenServer treats the two NICs as one logical connection.

Note: While NIC bonding can provide load balancing for traffic from multiple VMs, it cannot provide a single VM with the throughput of two NICs.

The illustration that follows shows how the cables and network configuration for the bonded NICs have to match.



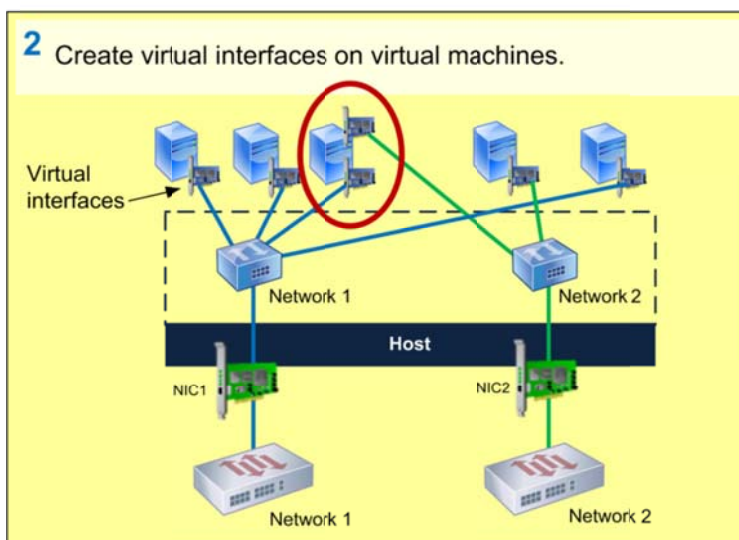
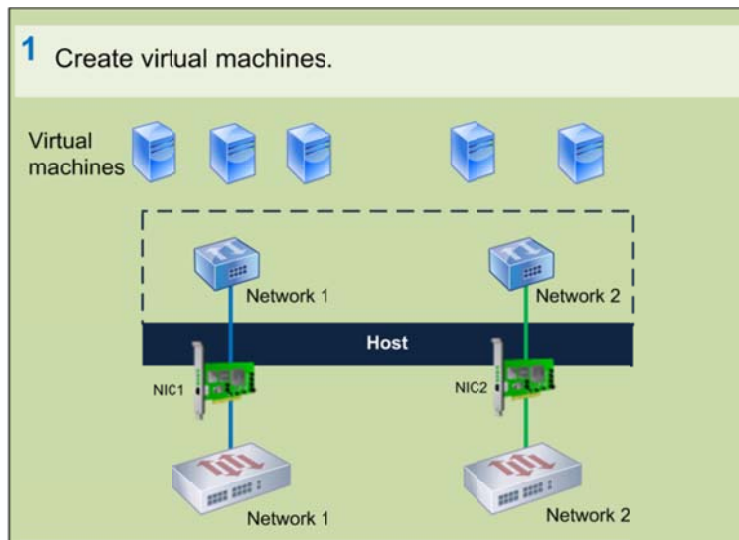
This illustration shows how two NICs in a bonded pair use the same network settings, as represented by the networks in each host. The NICs in the bonds connect to different switches for redundancy.

Note: For more information about bonds, see “Considering NIC Bonding” on page 56.

Connecting a VM to a Network using Virtual Interfaces

Virtual machines connect to a network through a virtual interface on that particular network. XenServer sends the VM's traffic through the target network's associated NIC. By default, when you create a VM in XenCenter, XenServer creates a virtual interface connecting the VM to Network 0. This configuration lets VMs connect to an external network through the NIC attached to Network 0.

You need a virtual interface on a VM for each separate physical network to which you want to connect it. In environments that connect to only one physical network, the virtual interface XenCenter creates by default when you create a VM may be sufficient for your needs. However, if you need a VM to connect to multiple physical networks, you must create a virtual interface for each one of those networks.



This illustration shows how VMs require a virtual interface for each physical network to which they need to connect.



Some additional points about virtual interfaces:

- Most, but not all, VMs have at least one virtual interface. (If an administrator accesses a VM only through XenCenter, the VM does not need a virtual interface.)
- Each virtual interface must have a “virtual” MAC address. You can configure XenServer to generate these automatically for you (recommended) or specify them manually.
- When you create a network in XenCenter, you can specify if you want XenCenter to create a new virtual interface for that network automatically, whenever you create a VM.
- Unlike for the physical and infrastructure layers, the networking configurations on VMs do not need to match other VMs in the pool.

Note: To determine which VM is associated with a virtual interface, see CTX122520 -- [How to Find which Virtual Network Interface is Assigned to a Virtual Machine in XenServer](#).

Understanding Virtual MAC Addressing

Just like NICs in the physical world, each virtual interface must have its own (virtual) MAC address. When you create a virtual interface, you can either specify a MAC address manually or let XenServer generate one for you.

When XenServer generates MAC addresses automatically, it generates *locally administered addresses*. Locally administered addresses are addresses assigned to devices by a user, which typically lack manufacturer-specific encoding. As a result, they do not contain a manufacturer-specific *Organizationally Unique Identifier* (OUI). Typically, manufacturers “burn-in” MAC addresses in which the first three octets indicate which company manufactured the device.

This means that the MAC addresses XenServer generates will not clash with addresses from *hardware* devices on your network.

XenServer generates a MAC addresses at random based on the random seed in the *VM.other-config:mac-seed* parameter of the VM and the device number of the virtual interface (a sequence number for the VIF: 0...6).

A particular combination of a MAC seed and device number always results in the same MAC address. Consequently, if you remove a virtual interface from a VM and recreate it later, the new virtual interface typically gets the same MAC as before.

XenServer preserves MAC addresses when migrating VMs. However, when you copy or clone VMs, the VM receives a new random MAC address seed and the virtual interfaces get new MAC addresses based on that seed.

Tip: To obtain the MAC address of a XenServer VM in XenCenter, select the VM’s **Network** tab, select the virtual interface, and click **Properties**.

Segregating VM Traffic from Management and Storage Traffic

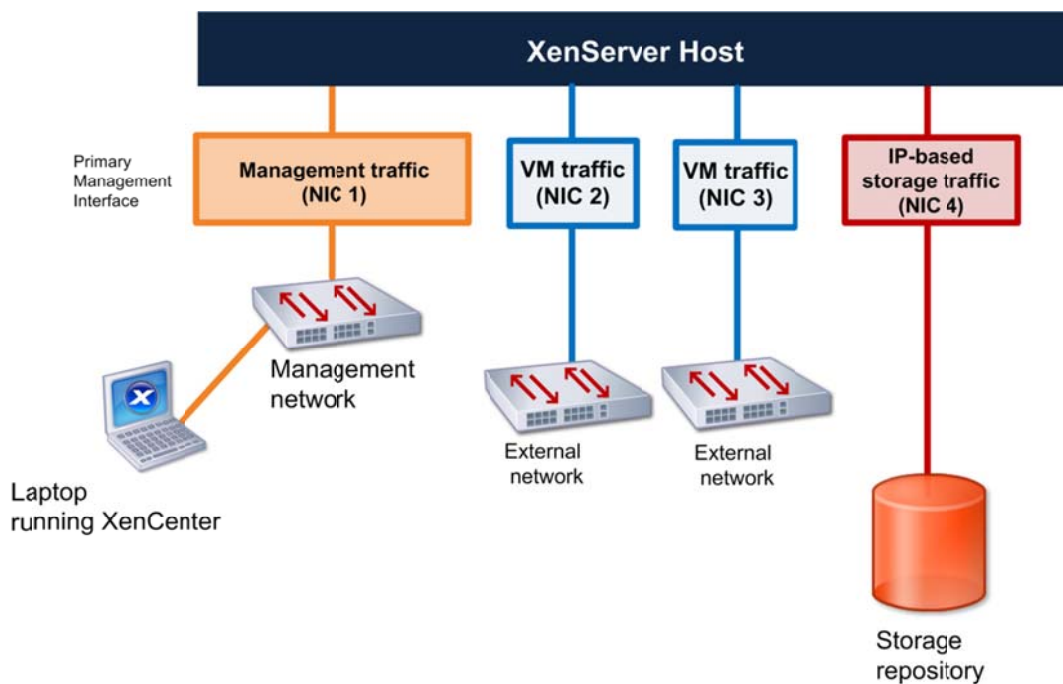
You can separate each type of traffic –VM, storage, and management traffic – onto its own network for either security or performance reasons.

For most environments, Citrix recommends segregating VM traffic from management traffic as the best practice. Not only does it increase the security of the management network, it can improve performance by reducing competition between traffic types for network resources, reducing potential collisions, and reducing the load on the primary management interface.

There are a variety of ways in which you can separate traffic, including:

- Separating all types of traffic from each other. For example, putting the virtual machines on a network not used for storage or management traffic.
- Separating the management traffic from the VM and storage traffic.

However, VMs will only use a NIC for VM traffic if they have a virtual interface on the same network as the NIC. The illustration that follows shows the best practice example of how you might separate traffic.



This illustration shows how NICs that are not designated for management or storage traffic only carry VM traffic.

While separating traffic is a best practice in larger environments, it is not an absolute requirement for all environments. In smaller environments, you may want to configure VMs to send their traffic on the management network. However, Citrix recommends evaluating the performance of this configuration regularly.

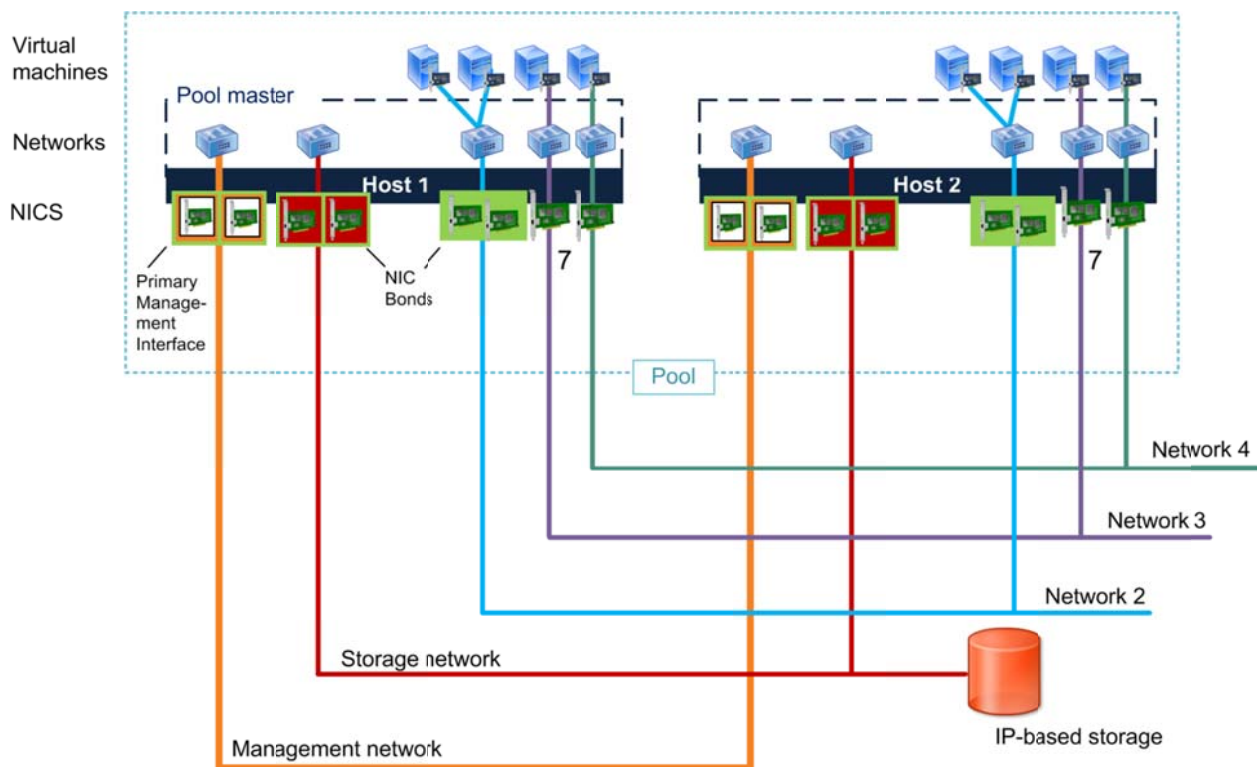
The scenarios that follow illustrate both of these concepts: separating traffic and sending traffic over NICs shared by multiple networks.

Scenario 1: Segregating Traffic

In this scenario, an administrator wants a dedicated network for management and storage traffic. To do this, the administrator:

- Attached the network cables coming from the NICs to a switch for a network to be used for VM traffic, which is physically isolated from the storage and management networks
- Created virtual interfaces on the same networks as the NICs

The illustration that follows shows these segregated networks.



This logical illustration shows segregated guest, storage, and management networks. In this scenario, all the VMs using network 2 can communicate with each other because they are configured to use the same (corresponding) NIC bond on their respective hosts and that bond connects to the same physical network. Likewise, the two VMs connected to network 3 can communicate with each since the corresponding NIC 7 on each host connects to the same physical switch.

As shown in previous illustration, not all NICs have virtual interfaces associated with them. If you do not configure a virtual interface connecting to the management network, the management NIC becomes dedicated for management traffic. For example, in the previous illustration there are NICs



connected to the management and storage networks that do not have corresponding virtual interfaces.

Note: Citrix does not recommend assigning IP addresses (that is, creating management interfaces) for each NIC on your host. Ideally, Citrix does not recommend using any NICs with IP addresses assigned to them for VM traffic.

Scenario 2: Using the Management Network for VM Traffic

In environments with minimal security requirements, you can configure VMs to share the management or storage networks.

In this example, the organization uses the management network for two purposes:

- XenCenter can connect to the management network through the primary management interface on the pool master. This is because of the IP address on that NIC. Likewise, hosts and other components, such as Workload Balancing, can use the connection to communicate with XenServer.

Note: XenCenter only communicates with the pool master and not any member servers. Specifically, XenCenter only connects to the IP address of the master's primary management interface.

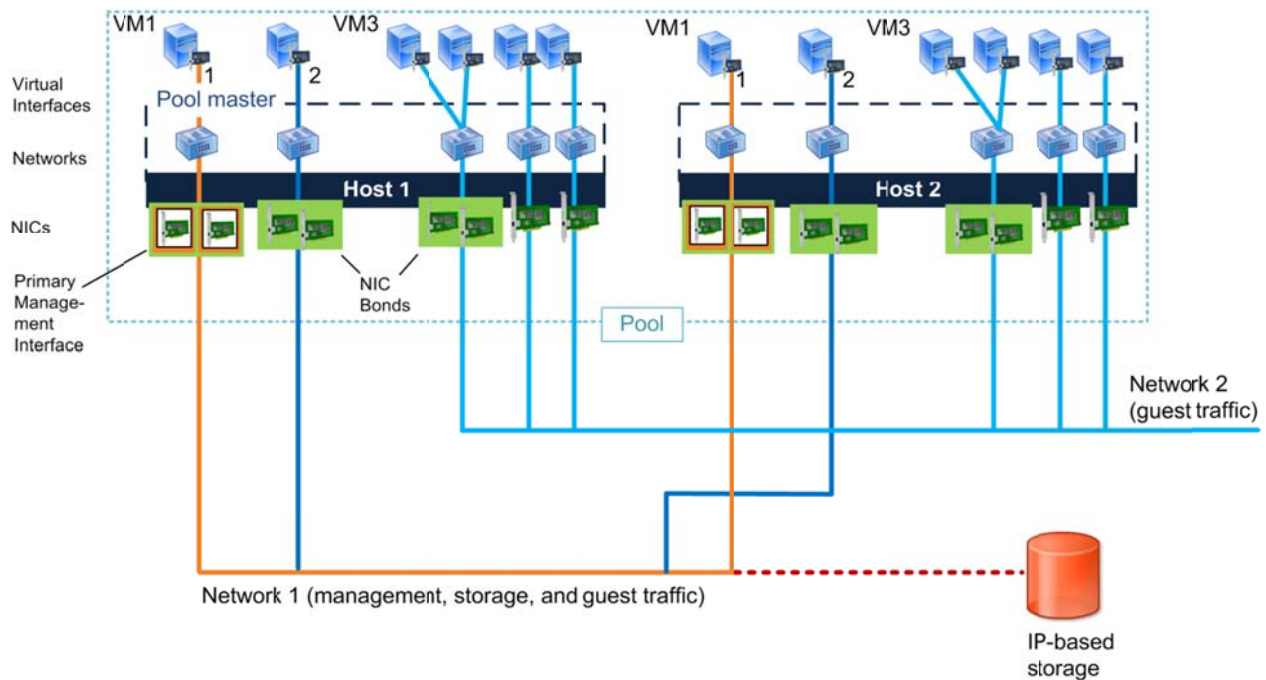
- VM traffic is also sent on this management network. This is the default configuration and requires no changes. To revert to this configuration, create a virtual interface on the VM and specify the VM network that is sharing the management network.

This configuration lets (1) XenServer use the NIC configured as the primary management interface to communicate with other hosts and (2) VMs transparently forward guest traffic onto that network and back.

However, this configuration has security implications. Workstations hosting XenCenter and XenServer hosts using this management network can communicate with each other because they are on the same network. This makes the management network, which ultimately manages the hardware layer and controls the hypervisors themselves, vulnerable to any attacks originating from the VMs. For example, if the VMs host Web servers, any successful attacks originating from outside the organization can potentially penetrate your entire virtual infrastructure – or all infrastructure on the targeted pool.

In contrast, scenario 1 on page 28 separates the VM traffic from the management network, which confines any successful external attacks to the guest network.

The following illustration shows some VMs sending their VM traffic over the management network.



This logical illustration shows how the administrator configured the virtual interfaces on VM 1 and VM 3 to send their traffic across the management network.

Note: Virtual interfaces appear differently in Linux and Windows VMs:

- In a Windows VM, the initial Windows installation has an emulated network device that uses a built-in driver.
- In a Linux VM, the NIC appears as a standard Linux network device and uses the high-speed Xen paravirtualized network driver.

After you install the XenServer Tools (for Windows guests), Windows also uses high-speed paravirtualized network drivers.

Scenario 3: Isolating VM Traffic on a Private Network

You might have specific types of workloads that require isolation. For example, in environments with technically savvy workers, you might not want servers with confidential employee data on the same network as regular VM traffic. XenServer lets you segregate traffic by creating two types of private networks: single-server private networks and cross-server private networks.

Private networks do not have an uplink or a physical NIC. Private networks connect VMs on the same XenServer host or the same resource pool. In a private network, VMs can only communicate with VMs on the same switch on the same host. In the case of cross-server private networks, VMs can only communicate with VMs on the same vSwitch.



Essentially, a private network functions like an isolated local area network that is local to either a host or a group of hosts (pool). This results in higher speed networks since responses between VMs are based on the storage speed and not limited by the network bandwidth or bottlenecks.

Due to the speed, lab machines and test environments are a good use case for private networks. Creating private networks might also be desirable for these reasons:

- **Security.** Single-server and cross-server private networks can let you isolate VMs from other network traffic (almost like creating a virtual “stove pipe”). Private networks and cross-server private networks are completely isolated from regular network traffic. VMs outside of the private network cannot sniff or inject traffic into the network, even if both sets of VMs are on the same physical server and the virtual interfaces on both sets of VMs transmit traffic across virtual interfaces connected to a network on the same underlying NIC.
- **Faster traffic for connections between VMs on the same host.** Because VMs do not need to interact with regular network and switches, they can transmit traffic faster to each other.

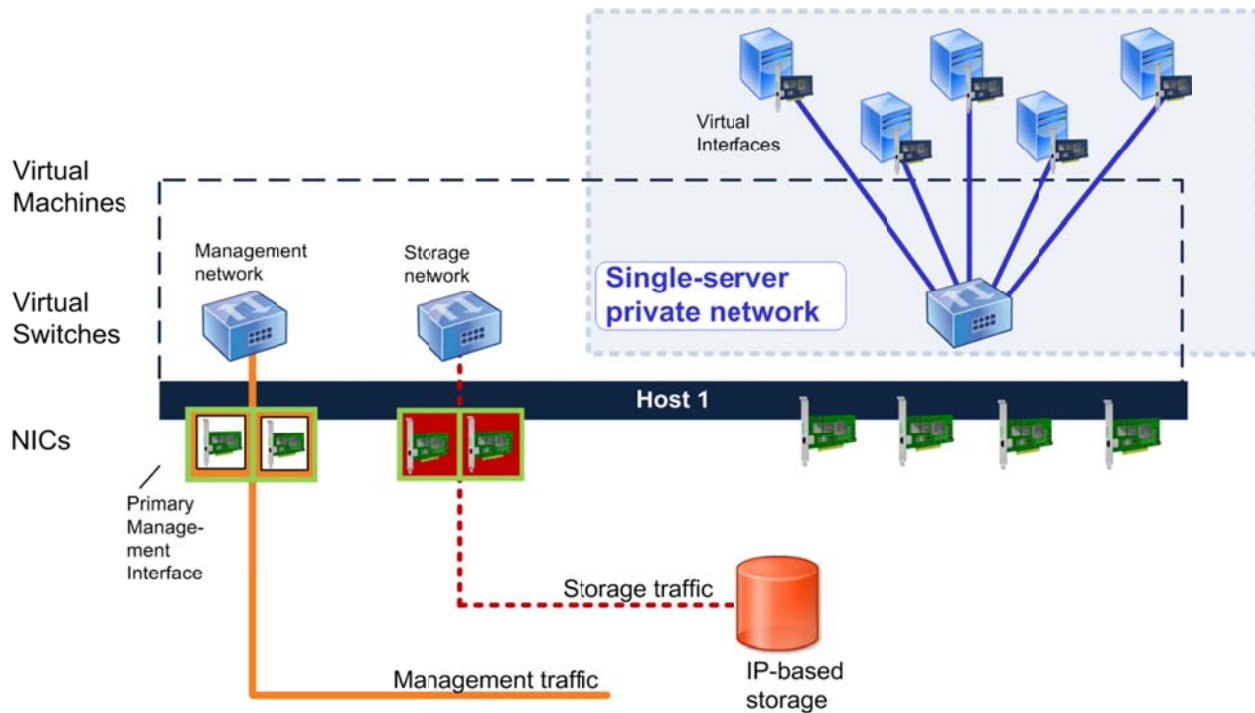
Private networks provide connectivity only between VMs on a given XenServer host and do not have a connection to the outside world. Networks with a NIC (PIF) association are considered external: they provide a bridge between virtual interfaces and the NIC connected to the network, enabling connectivity to resources available through the NIC.

Note: In previous XenServer releases, single-server private networks were known as internal networks.

Scenario A: Isolating VM Traffic on One Host

If you have some VMs on one host that you do not want on your organization’s network, you can create a *single-server private network*. This is an internal network that has no association with a physical network interface. It only connects the virtual machines on the host and has no connection to the outside world.

The illustration that follows shows a private network configured on one host.



This illustration shows how the virtual interfaces on the VMs are on the single-server private network. This network does not have any connect to any NICs since all traffic is sent inside the XenServer host.

To create a single-server private network that is isolated from the external network, you

1. Create a single-server private network in XenCenter.
 In XenCenter, select the host in the Resource pane. Click the **Network** tab. Click **Add Network** and then select **Single-Server Private Network**.

Unlike when you create external networks, XenCenter does not prompt you to specify a NIC when you create private networks. This is because private networks do not require a NIC for connectivity.

2. Create a virtual interface on each VM that specifies the new private network.

If you want to isolate the VMs' traffic completely, if necessary, remove any virtual interfaces on the VMs that are on an external network.

Note: To create cross-server private networks, see CTX130423 - [Citrix XenServer 6.0 vSwitch Controller User Guide](#).

Scenario B: Isolating VM Traffic on Cross-server Private Networks

Cross-server private networks are similar to single-server private network except cross-server private networks let VMs on different hosts communicate with each other. Cross-server private networks combine the isolation properties of a single-server private network with the additional ability to span



hosts across a resource pool. This combination allows the use of VM agility features, such as XenMotion (live migration) and Workload Balancing (WLB), for VMs connected to those networks.

Cross-server private networks are completely isolated. VMs that are not connected to this type of private network cannot sniff or inject traffic into the network, even when the VMs share a host that has virtual interfaces connected to two different networks that use the same NIC.

While VLANs provide similar functionality, cross-server private networks provide isolation without requiring physical-switch configuration.

Cross-server private networks provide the following benefits:

- The isolation properties of single-server private networks
- The ability to span a resource pool, enabling VMs connected to a private network to live on multiple hosts within the same pool

Because cross-server private networks require a NIC with an IP address, to configure these networks you must create a management interface. Cross-server private networks can use any management interface as the underlying network transport. However, if you choose to put cross-server-private network traffic on a second management interface, this second management interface must be on a separate subnet.

To create a cross-server private network, the following requirements must be met:

- All hosts in the pool must be using XenServer 5.6 Feature Pack 1 or greater
- All hosts in the pool must be using the vSwitch for networking
- The pool must have a vSwitch Controller configured
- The cross-server private network must be created on IP-enabled NICs (that is, a management interface)

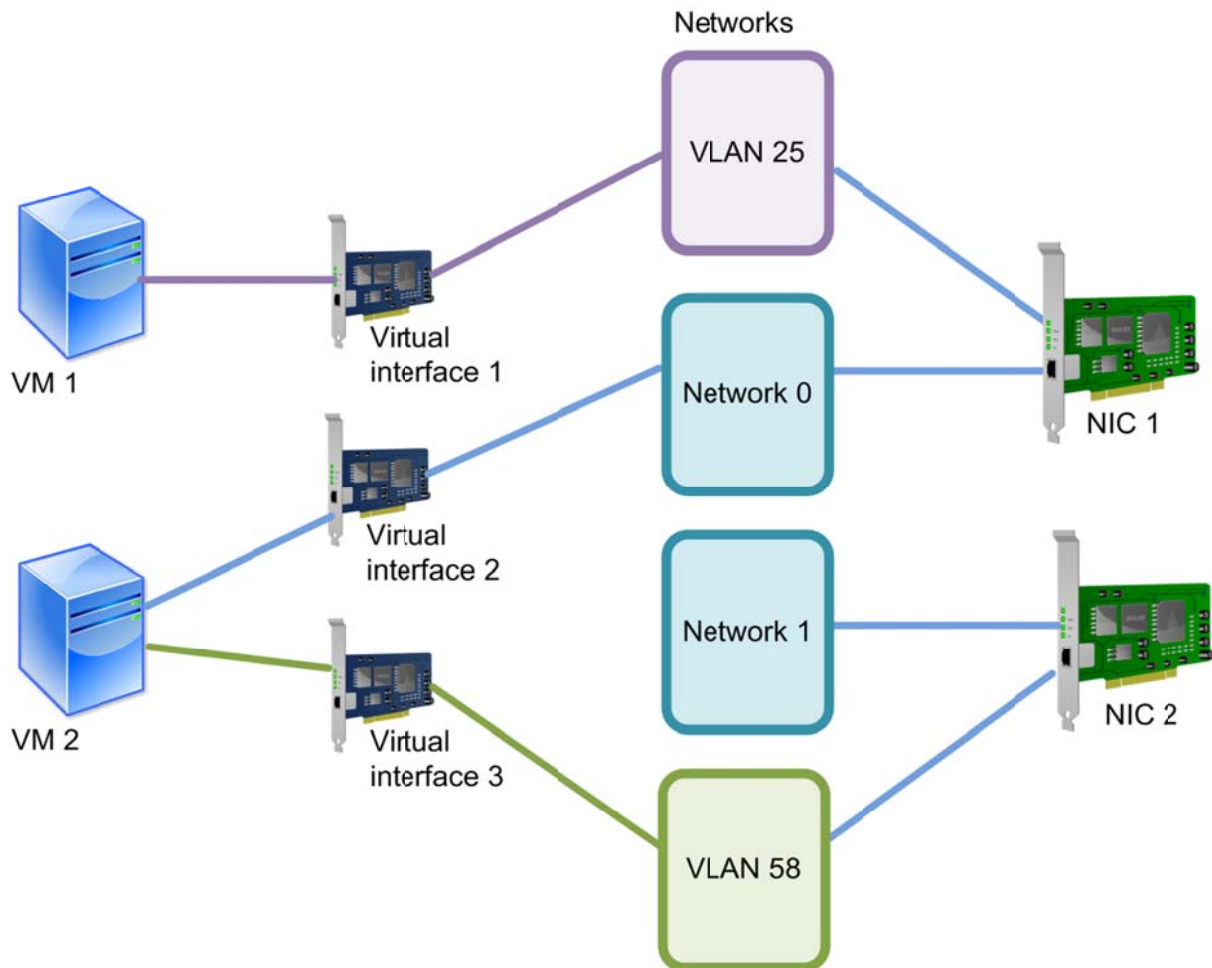
To create a cross-server private network using XenCenter

1. In XenCenter, select the pool where you want to create the network in the Resources pane.
2. On the first page of the **New Network** wizard, select **Cross-Server Private Network** and click **Next**.
3. Enter a name and description for the new network, and click **Next**.
4. Do one or more of the following:
 - To automatically add the new network to any new virtual machines created using the New VM wizard, select the check box.
 - To use jumbo frames, set the Maximum Transmission Unit (MTU) to a value between 1500 to 9216.
5. Click **Finish** to create the new network.

Scenario 4: Connecting VMs to Multiple Linked VLANs

Many organizations today configure VLANs to logically separate their physical networks for either performance or security reasons. If your organization has VLANs, you might want to connect your VMs to one or more VLANs on your network.

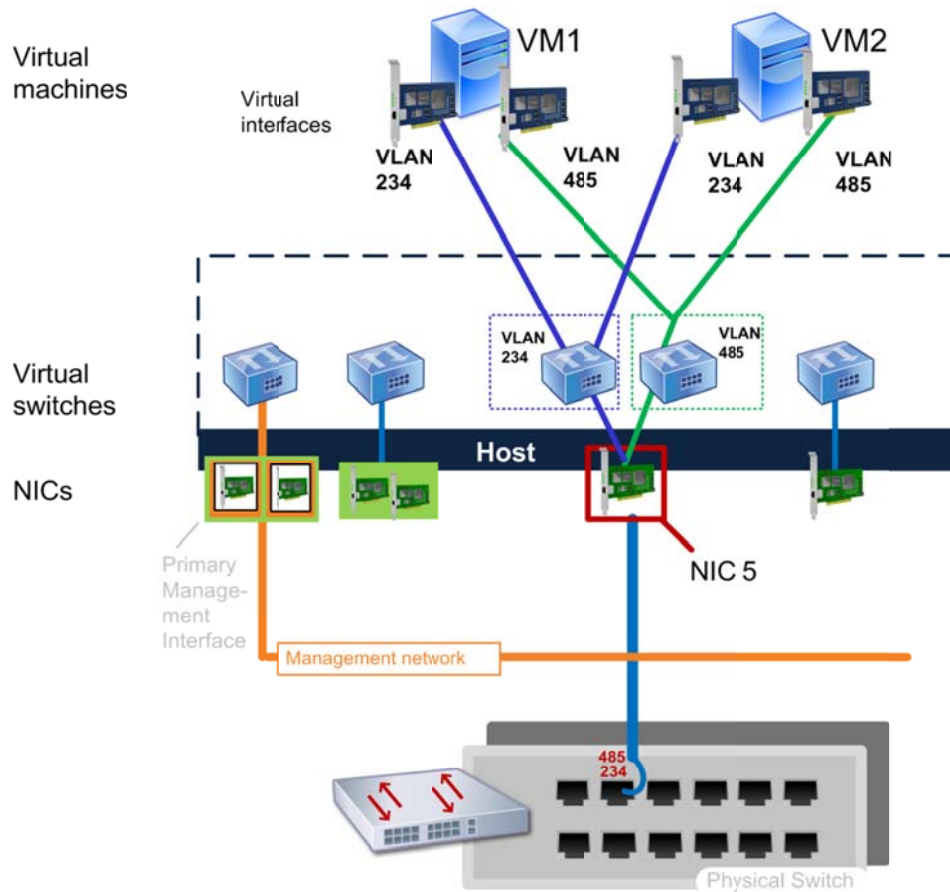
To connect a VM to a VLAN, you must create a network for the VLAN and then connect the VM to that network. To perform this configuration, you create a separate external network for each VLAN and then create a virtual interface on the VM for each of these networks.



This illustration shows how VMs require a separate virtual interface for each network to which you want to connect them, including VLANs. In this example, VM 2 connects to Network 0 through Virtual Interface 2 and to VLAN 58 through Virtual Interface 3. As shown by VM1 and NIC1, multiple networks can connect out through one NIC.

While trunk lines from the physical switch can contain multiple 802.1q VLANs, XenServer does not let you combine multiple VLANs in one XenServer network. This means that to let a VM connect to multiple VLANs you must either (a) create a separate network in XenServer for each VLAN or (b) create a XenServer network for a VLAN that can access all of the desired VLANs.

In the illustration that follows, the VMs connect to a VLAN through a trunked switch port.



This illustration shows how VMs on the host connect to an external network that the administrator configured to connect to VLAN 485 and VLAN 234. To achieve this, the administrator created an external network that uses NIC 5 to connect to a trunked switch port that includes VLAN 485 and a second external network that also uses NIC 5 to connect to VLAN 234. The administrator ran a cable from the VLAN trunk port to NIC 5.

Connecting a VM to a VLAN requires that you:

1. Create a physical connection between the corresponding NIC on each host and the VLAN trunk port for that VLAN on the switch.

For example, if you connect NIC 7 on the XenServer pool master to a VLAN trunk port on the switch with access to VLAN 485, you must run a cable from NIC 7 on all other hosts in the pool to a similarly configured VLAN trunk port on the same switch, which can access VLAN 485.

2. Enable XenServer to connect to a specific VLAN on the switch by creating an external network specifying that VLAN tag.

This means creating an external network on the XenServer pool master and specifying the VLAN tag when you create the network.

In XenCenter, select the pool (<*your-pool-name*>) in the **Resource** pane, click the **Network** tab, and click the **Add Network** button. In the New Network wizard, select **External Network**. On the **Location** page, specify the NIC you physically connected to the switch and enter the VLAN tag for the VLAN in the **VLAN** box.

In the XenServer CLI, you can use the **pool-vlan-create** xe command to create the VLAN on all hosts in a resource pool. For more information, see the *XenServer Administrator's Guide*.

After you create the network for the VLAN on the pool master, XenServer configures the NICs on all the other hosts so that the corresponding NIC on each host

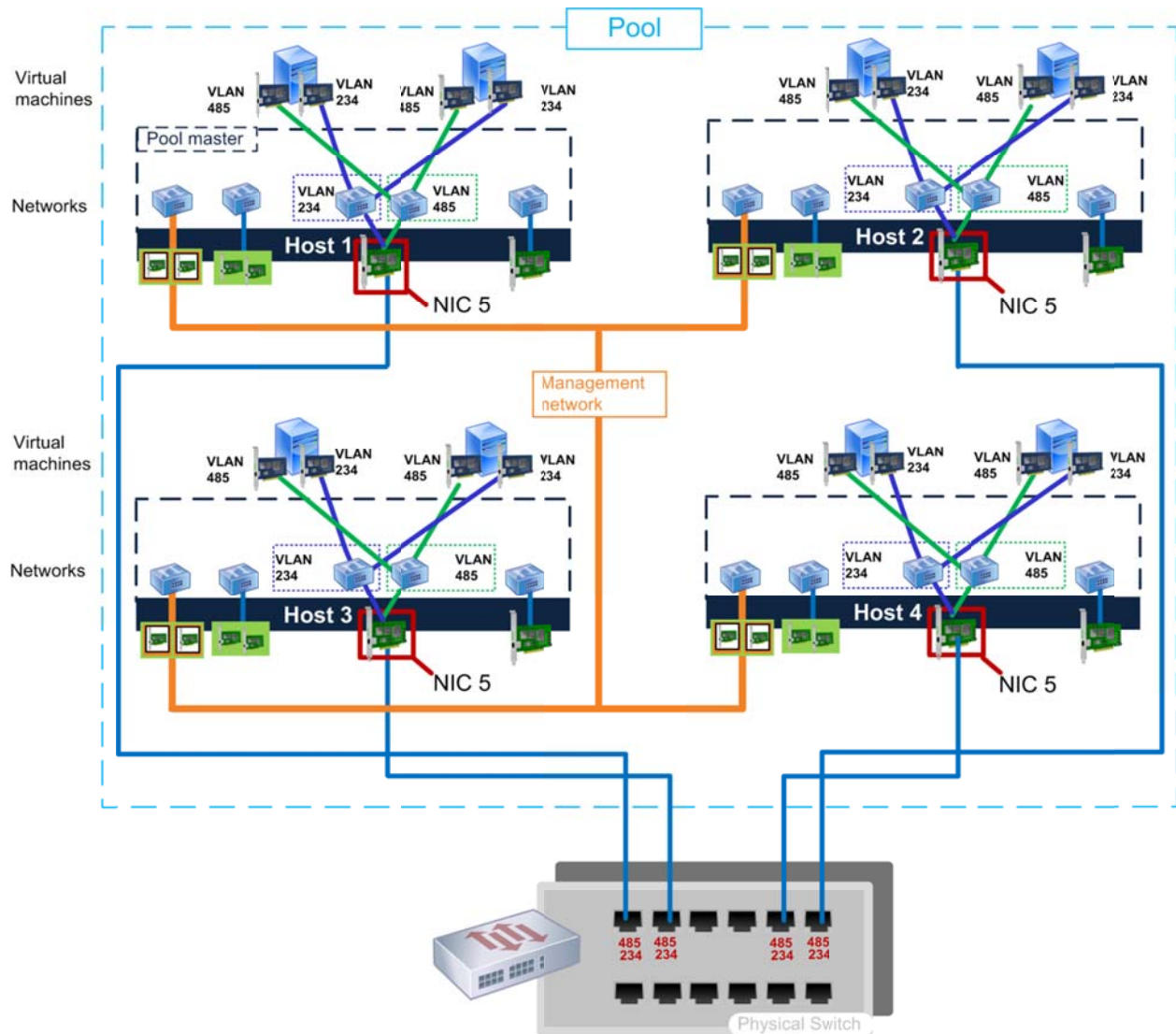
Note: The numbers of VLAN tags must be between 0 to 4094.

3. Connecting the appropriate VMs to the VLAN by configuring a virtual interface that points to that network on each VM you want to be able to connect to the VLAN.

In XenCenter, this is done by selecting the VM in the Resource pane, clicking the **Network** tab, and clicking **Add Interface** and then specifying the VLAN network when you create the interface.

Again, because networking is a pool-level feature, if you connect one host to a VLAN, you must connect all hosts in the pool to the VLAN. This means that you must physically connect the corresponding NIC on each host to the VLAN port on the switch.

In the illustration that follows the VMs on multiple hosts in a pool connect to a VLAN through a trunked switch port.



This illustration shows how, because XenServer automatically synchronizes the network settings in pools so that they match, NIC 7 on all hosts in the pool will be configured with the same network and VLAN settings as NIC 7 on the pool master. However, for the VMs on the member servers to be able to connect to the VLAN, the administrator must also physically connect NIC 7 on each host to a trunk port on the switch that can access VLAN 485.

Before configuring a VLAN, ensure the switch on your VLAN network is configured as follows:

- The port on the switch connected to each XenServer host must be configured as trunk port.
- The port on the switch must be configured for 802.1q encapsulation.
- Port security cannot be set on the trunk port.
- The port designated as trunk should be assigned a native VLAN; use 1 as default.

XenServer lets you create multiple networks and VLAN networks on the same NIC. XenServer does not limit the number of VLANs you can connect to VMs. Instead, the limit comes from the



802.1q standard is 4096. You add an external network for each VLAN to the host and then connect the VMs to the VLANs by specifying that network in the VM's virtual interface.

Note: If a Native VLAN is used on the switch trunk port, then you cannot assign that VLAN number to a VM on the XenServer.

For an example of a tested working model of a VLAN configuration, see CTX123489 -- [XenServer VLAN Networking](#). For more information about configuring VLANs on your switch and 802.1q support, see the documentation for your switches.

Tip: To verify that you have configured the XenServer host to communicate across the correct network, you can use the packet sniffing software included with your NICs to capture and display the VLAN tags that are transmitted across the switch to the XenServer.

Note: Although the 802.1q standard limits the number of VLANs XenServer supports, XenServer does not limit the number of XenServer networks you can configure for a NIC.

Understanding the Impact of Numerous Networks in a Pool

Having numerous connections to VLANs (for example, 100s) configured on a host creates an additional load on the Control Domain, which frequently results in reduced network performance as described.

Having numerous VLANs can also impact your host, pool, and VMs performance in the following ways:

- VM performance may degrade.
- VM network service may degrade. However, this can be due to many factors.
- Numerous VLANs can slow down certain host (XenAPI) operations, such as adding and removing networks.

In addition, various management and administration functions can become slower when there are numerous networks on pools. For example, actions like the following may take longer: joining a new host to a pool, rebooting a host; rendering charts in the Performance tab in XenCenter.

Creating VLANs on Bonded Networks

XenServer supports connecting to VLANs from bonded NICs. To do so, do the following:

1. Bond the two NICs together. After you have done so, the NIC bond appears as a bonded network in XenCenter.
2. In XenCenter, for example, create an **External Network** specifying the following:
 - a) The VLAN's tag
 - b) The NIC bond as the NIC



You might want to name this external network the same name as the VLAN (for example, VLAN 25).

3. When you create the virtual interface for the VM, specify the external network with the VLAN tag as the network.

Creating VLANs on the Primary Management Interface

You can have a single VLAN on the primary management interface, and this VLAN can be on an access port. If you want to use a trunk, either you define a default VLAN on that trunk and the management interface can use that or you make the port a full access port.

XenServer does not support having a VLAN trunk port on the primary management interface.

Chapter 4: Specifying Networking Requirements

This chapter provides information about the following:

- XenServer networking requirements and support
- A suggested process for defining your communication and hardware requirements

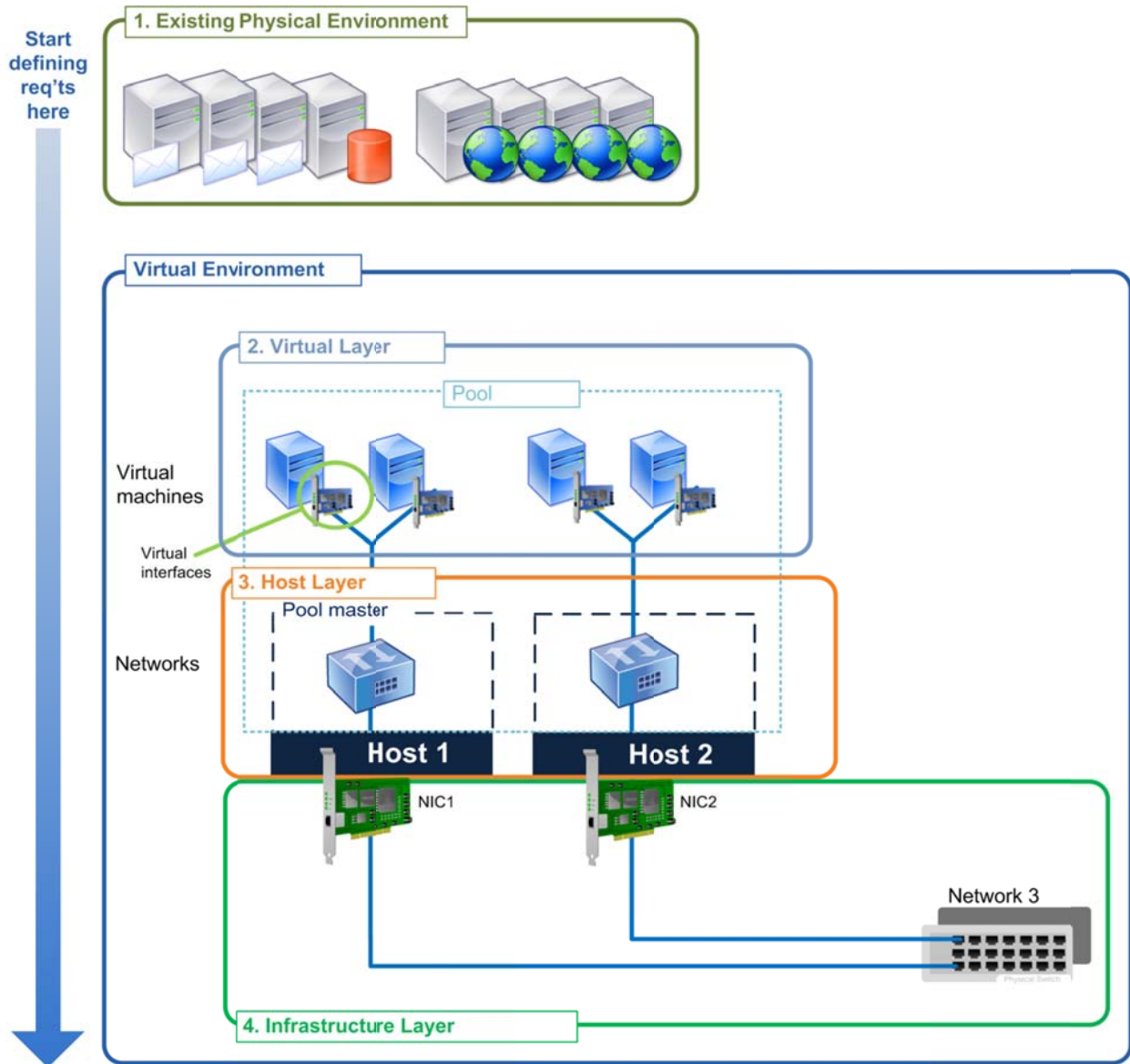
Overview

This chapter helps you map your existing networking requirements onto XenServer features and includes principles to note while defining your XenServer networking requirements. It also provides information about XenServer networking requirements and supported configurations.

This chapter includes tables that map common requirements and configuration scenarios onto XenServer features and indicate what topics to read for more information. The topics covered in the tables include guidance about defining basic requirements at the VM level and at the host level.

The tables are structured so that you begin your evaluation by considering your existing workloads (for example, in your physical environment) since this will indicate your VM's connectivity requirements. After determining your VM's communication requirements, define how you want to group VMs together and the host's networking hardware.

The following illustration shows a suggested sequence or “direction” in which to consider networking requirements: from the physical environment down to the layers of the virtual environment.



This diagram presents a possible way of defining networking requirements for a XenServer pool. It suggests that you begin by examining your physical environment and then consider your networking requirements by beginning at the VM workload level, defining networks required at the host level, and then defining your physical infrastructure requirements, such as NIC and switch configuration.

Introduction

The primary factor that drives your networking requirements is the connectivity needs of the pool's VMs and their workloads. In some cases, you may choose to group VMs in pools according to the networking requirements of their workloads. This could either be due to the:

- Workloads' networking hardware requirements since all hosts in a pool must use the same networking hardware



- Workloads' networking configuration since sometimes you might want to create pools based on common networking requirements so as to reduce the amount of networking configuration you must perform

As a result, it helps to know the workloads you want to virtualize and their networking requirements before you begin configuring networking. Likewise, you should know the approximate the size of your pool (that is, the number of hosts).

Ideally, all networking configuration is performed before you put a pool into production. However, you do not need to configure all of the pools in your virtual environment when you begin configuring networking. Rather, you can configure one pool at a time. Although configuring networking before putting a pool into production is a best practice, you can add hosts and make networking changes at any time.

XenServer Networking Support and Requirements

This section provides information about the physical and logical networking configurations XenServer supports, such as the number of NICs or networks supported. It also provides information about where to find a list of supported networking hardware.

When you are defining networking requirements for a pool, it is important to note that all pooled hosts should have the same number and model of NICs, same XenServer networks, and physical cabling configuration. Because XenServer assumes all network settings in a pool match, it automatically propagates any changes you make to network settings on one host to all other hosts in the pool.

XenServer Supported Configurations

XenServer supports the following networking configurations:

- Up to 16 physical network interfaces (or up to 8 pairs of bonded network interfaces) per XenServer host.
- Up to 7 virtual interfaces per VM.
- Active-active and active-passive bonding modes are supported. DMP and RDAC MPP multipath handlers are also supported.
- Four different types of networks: external, cross-server private, single-server private, and VLANs.
 - There is no Citrix-imposed preset limit on the number of VLANs.
- SR-IOV provided the NIC used meets the support requirements in the [Citrix XenServer Hardware Compatibility List](#).



XenServer Networking Hardware Requirements

Citrix Technical Support only provides support for hardware, including NICs, on the [Citrix XenServer Hardware Compatibility List](#). While it may be possible to use different hardware, it is not recommended. If you do not see your hardware on the list, you can use the self-certification kits at the [Citrix XenServer Hardware Compatibility List](#) web page and certify it yourself by running the kit and submitting your results.

Important: Do not configure any networking settings until you have added all the hosts to the pool and you finish physically connecting each host to the appropriate switch ports. Then, proceed to configure your XenServer network settings by starting with (and configuring only) the pool master.

Defining Your Networking Requirements

Defining network requirements for your XenServer environment is a multi-stage process. Two factors significantly influence your network design and requirements:

1. Certain workloads may have specific network connectivity or performance requirements.
2. While VMs in a pool can connect to different networks, the network configurations and hardware on each host in a pool must match.

As a result, you might find it easier to define your workloads' network requirements before you determine in which servers and pools to host your VMs.

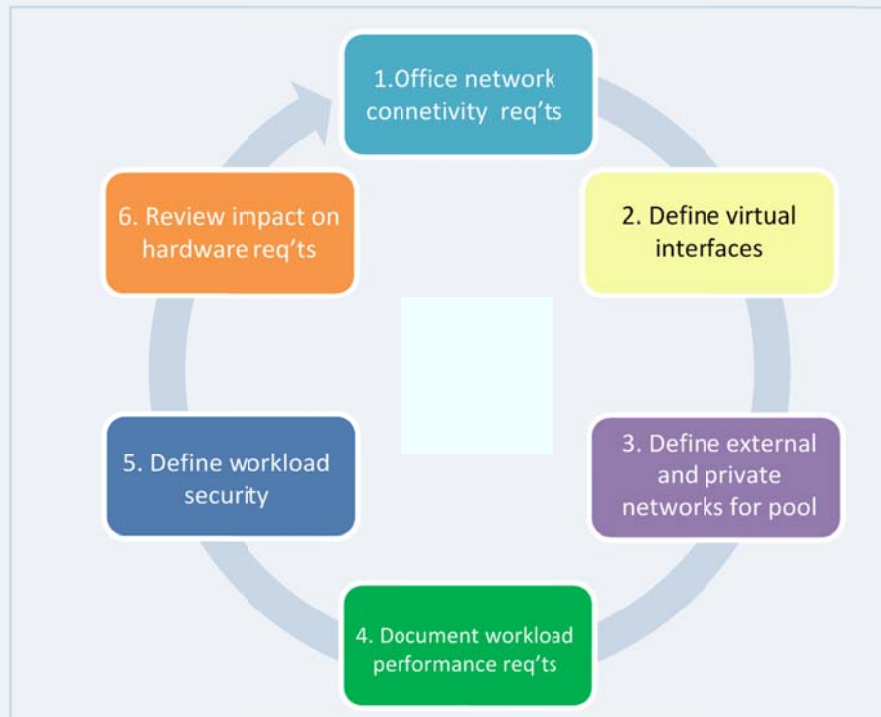
This section provides guidance about how to begin determining the communication requirements for your workloads, how to evaluate your existing networking configuration, and how to review your initial pool design against the network requirements of its workloads.

Considering Workload Communication Requirements

To determine a host's networking configuration, start evaluating requirements at the lowest level (that is, the VM/workload) and then work your way up to your organization's physical network. The networking requirements for your workloads can ultimately impact your pool design and which workloads you decide to group together.

To connect a VM to an external network, such as a VLAN, you might determine the VM's networking requirements by using a process like the one in the following illustration.

Defining Networking Requirements for a Workload



This diagram illustrates a general process you might use when defining networking requirements at the workload level.

Determine the workload’s communication requirements (for example, does a domain controller need access to a specific VLAN or a database server need access to a specific storage device?). To do so, consider the factors listed in the following table:

Factor	Action, Notes
Determine if the workload has any specific performance requirements that might change its host’s hardware design.	<p>Some workloads might require specific NICs due to performance requirements.</p> <p>An option for optimizing virtualized Provisioning Services servers is SR-IOV – see “Virtualizing the Provisioning Services Server” on page 81.</p> <p>See the XenServer Hardware Compatibility List for supported NICs.</p>
Consider the redundancy	For information about NIC bonding, see “Designing Network

Factor	Action, Notes
requirements for the workload. Is the workload mission critical?	<p>Redundancy” on page 56.</p> <p>For redundancy, you can also look at solutions by Citrix partners, such as Marathon, which provide multi-CPU full-compute failure fault tolerance.</p>

Evaluating Your Current Network Configuration

When designing your XenServer network configuration, consider the following aspects of your current physical networking configuration:

Factor	More information
Security and Isolation	
Consider your workload’s communication requirements and how you will connect its VM to the devices and/or network locations in your external network.	<p>VMs connect to networks using virtual interfaces as described in “Chapter 3: Sample Networking Scenario.”</p> <p>If your VM must connect to a specific VLAN that is part of a trunk port, you must create a specific external network associated with the VLAN. For more information, see “Scenario 3: Isolating VM Traffic on a Private Network” on page 32.</p>
Does the workload need to be isolated from other traffic?	<p>Configure either a:</p> <p>VLAN subnet or a trunk port</p> <p>or a</p> <p>Cross-server private network as described on page 32.</p> <ul style="list-style-type: none"> • Additionally, you can use the Distributed Virtual Switch controller to isolate VMs by blocking specific machines or IP addresses from communicating with each other. • Cross-Server private network as described in “Scenario 3:

Factor	More information
	Isolating VM Traffic on a Private Network” on page 30.
Will you need to perform live migration for this workload?	<p>For pools with VMs that use large amounts of memory or pools that will have frequent migrations, consider using the primary management physical interface just for live migration and management tasks. In this scenario, all VM, storage, and other traffic should be placed on separate physical interfaces. See “Segregating VM Traffic from Management and Storage Traffic” on page 27.</p> <p>Live migration requires shared storage.</p>
Configuration	
Does this workload require you enable promiscuous mode?	See “Enabling Promiscuous Mode for Traffic Monitoring” on page 68.
Do you want XenServer to automatically generate MAC addresses for the VM hosting this workload or do you want to assign the VM a static IP address?	Citrix generally recommends letting XenServer automatically generate MAC addresses. See “Understanding Virtual MAC Addressing” on page 26.

Determining Host Networking Requirements

Factor	More information
Does the workload need all of the bandwidth available or should it be deprioritized in favor of other workloads?	Set a quality of service (QoS) restriction on the virtual interface for the VM. See “Limiting Bandwidth Consumption for High Demand Workloads” on page 66.
Storage Requirements	
Does the storage device require an IP address on the host?	This requires configuring a (storage) management interface so you can assign an IP address to the NIC. For more information, see “Segregating VM Traffic from Management and Storage

Factor	More information
	"Traffic" on page 27 and "Chapter 6: Designing Your Storage Network Configuration" on page 70.
Are you planning to use an iSCSI Host Bus Adapter (HBA)?	For a list of supported HBAs, see the Citrix XenServer Hardware Compatibility List .
Would the storage traffic benefit from configuring jumbo frame support?	<p>Currently, jumbo frames are only supported for storage networks with iSCSI HBAs and the vSwitch configured as the networking bridge. This means:</p> <ol style="list-style-type: none"> 1. If you want to configure jumbo frames for your storage traffic, you must use a storage device that can use an HBA, such as an iSCSI hardware SAN or a Fibre Channel SAN. 2. You must configure the vSwitch as the networking bridge. You can choose to configure the Distributed Virtual Switch solution or just configure the vSwitch as the bridge. See "Deciding to Use the Distributed Virtual Switch" on page 53. 3. You must configure end-to-end support for your jumbo frames, including switches and NICs that are compatible with them. For more information, see "Configuring Networks with Jumbo Frames" on page 78.
Do you need to change duplex settings on the host?	See CTX117568 -- How to Modify Network Speed and Duplexing .

Reviewing Initial Pool Design against Networking Requirements

After you have determined a possible pool design for a group of workloads, consider if any of the following will cause problems:

Factor	Action, Notes
Will the network hardware requirements for these workloads clash? Hardware in pools should	If a subset of workloads require more expensive NICs, such as ones that support jumbo frames or 10 gigabit Ethernet, do you want to purchase that networking hardware only for the servers

Factor	Action, Notes
match.	hosting those workloads? If so, you should group these workloads in the same pool(s). See “Impact of Pools on XenServer Networking” on page 15.

Considering Addressing Requirements

This section discusses IP addressing considerations.

Factor	Action, Notes
How do you want to configure IP addressing for the primary and (storage) management interfaces?	<p>Unless configured as a management interface, only the primary management interface requires an IP address: by default, all other NICs do not have an IP addresses.</p> <p>You can assign static IP addresses to each NIC in the host through XenCenter or the xsconsole.</p> <p>You specify the IP address for the primary management interface during XenServer Setup. You can specify that XenServer uses either static or dynamic IP addresses. However, assigning hosts static IP addresses is generally preferred. For more information, see “Networking Configuration after Installation” on page 14.</p> <p>IP-based storage requires configuring a (storage) management interface so you can assign an IP address to the NIC. For more information, see “Segregating VM Traffic from Management and Storage Traffic” on page 27 and “Chapter 6: Designing Your Storage Network Configuration” on page 70</p> <p>For information about setting a static IP address after Setup, see CTX116372 -- How to Assign a Static IP Address to a XenServer Host.</p>



Calculating the Number of Physical NICs per Host

All XenServer hosts in a pool should have the same number of NICs; however, this requirement is not strictly enforced when a XenServer host joins a pool.

Having the same physical networking configuration for XenServer hosts within a pool is important because all hosts in a pool share a common set of XenServer networks. The NICs on each host connect to pool-wide networks based on device name. For example, all XenServer hosts in a pool with an eth0 NIC will have a corresponding NIC plugged into the pool-wide Network 0 network. The same will be true for hosts with eth1 NICs and Network 1, as well as other NICs present in at least one XenServer host in the pool.

If one XenServer host has a different number of NICs than other hosts in the pool, issues can occur because not all pool networks will be valid for all pool hosts. For example, if host1 and host2 are in the same pool and host1 has four NICs while host2 only has two, only the networks connected to NICs corresponding to eth0 and eth1 are valid on host2. VMs on host1 with virtual interfaces connected to networks corresponding to eth2 and eth3 will not be able to migrate to host2.

The number of physical NICs you want on each host (and, consequently, the pool) depends on your required resiliency and connectivity. For an example of a reason you might want to use additional NICs to separate traffic, see “Segregating VM Traffic from Management and Storage Traffic” on page 27.

Although XenServer can be run with only one NIC on the host, Citrix recommends having at least two NICs on the host: one for VM traffic and one for management traffic. Other examples of how you could use NICs include:

- A best practice is to bond the NICs for the primary management interface and the one for VM traffic, which means deploying at least four NICs per host. If you are using IP-based storage, you might want six NICs.
- You may want to provide additional bandwidth for VM traffic by adding additional NICs to the host.
- If you are going to have shared storage, consider dedicating a physical network for your storage traffic. In this case, consider having at least one NIC or HBA, or ideally two in a bonded or multipathed configuration, dedicated to the storage traffic.
- If you are using Provisioning Services to stream disk images to VMs, you may want to dedicate a bond dedicated for the Provisioning Services server on the XenServer host.

Consider the factors in the following table to help determine the number of NICs you want in your hosts:

Factor	Action, Notes
Do you want to optimize your IP-based storage or provide redundancy?	<ul style="list-style-type: none"> • The best practice is to configure two NICs or iSCSI HBAs for storage traffic in a bonded or multipath setup. • For bonding, see: <ul style="list-style-type: none"> ○ “Creating Network Resiliency through Bonds” on page 23. ○ “Considering NIC Bonding” on page 56. • For multipathing, see: <ul style="list-style-type: none"> ○ CTX121364 -- Multipath Implementations in XenServer. ○ CTX118791 -- Multipathing Overview for XenServer 5.0. This article provides a good, albeit somewhat dated, overview of the UI-based multipathing configuration process.

For information about adding additional NICs to XenServer, see CTX121615 -- [How to Add Additional Physical NICs to XenServer](#).

For an example of how to calculate the number of NICs for XenDesktop-XenServer deployments, see the 22 February 2011 Citrix blog post, “[XenServer for XenDesktop - How many network cards do I need?](#)”

Calculating Bandwidth Requirements

Estimating how much network bandwidth your virtualized environment will need is a key part of ensuring good VM performance. Because all VMs on a host share the host’s bandwidth, providing the host with enough bandwidth is critical. Factors that affect bandwidth requirements include:

Factor	Action, Notes
Number of VMs	<ul style="list-style-type: none"> • Servers hosting more VMs may require more bandwidth, depending on the type of workload. See “Designing Networks for Performance” on page 63.
Type of workload and traffic	<ul style="list-style-type: none"> • Some server roles require more bandwidth. For example, servers sending a lot of traffic to storage devices or that have a lot of back up

Factor	Action, Notes
	<p>traffic.</p> <ul style="list-style-type: none"> Some operating systems have a lower impact on network performance. See “Designing Networks for Performance” on page 63.
Workload-specific bandwidth requirements	<ul style="list-style-type: none"> In some cases, you might need to constrain VMs to ensure sufficient bandwidth. For example, you might want to configure QoS policies on the NIC to specify a rate limit. See “Limiting Bandwidth Consumption for High Demand Workloads” on page 66.
Provisioning Services	<ul style="list-style-type: none"> A key concern in a Provisioning Services deployment, especially for large XenDesktop implementations, is the IOPS required for servers and target devices when the VMs boot. See CTX128645 -- Design Considerations for Virtualizing Provisioning Services.

For information about testing XenServer network performance, see “Testing XenServer Network Performance” on page 65.

Chapter 5: Designing XenServer Networks

This chapter provides an overview of the decisions you need to make when designing your XenServer networking configuration and contains information about the following:

- Deciding whether or not to implement Distributed Virtual Switching
- Designing your XenServer networking configuration for redundancy
- Isolating traffic
- Designing for performance
- Hardware and NIC specific configurations

Overview

At a high level, most XenServer network design decisions stem from a combination of three major design goals: the need for redundancy, performance, or isolation. For many organizations, these goals may overlap and are not necessarily mutually exclusive.

While considering these goals, keep in mind that the physical network configurations you create for one host should match those on all other hosts in the pool. Consequently, it helps to understand similarities between networking requirements for different workloads before you group workloads together in hosts and, ultimately, pools.

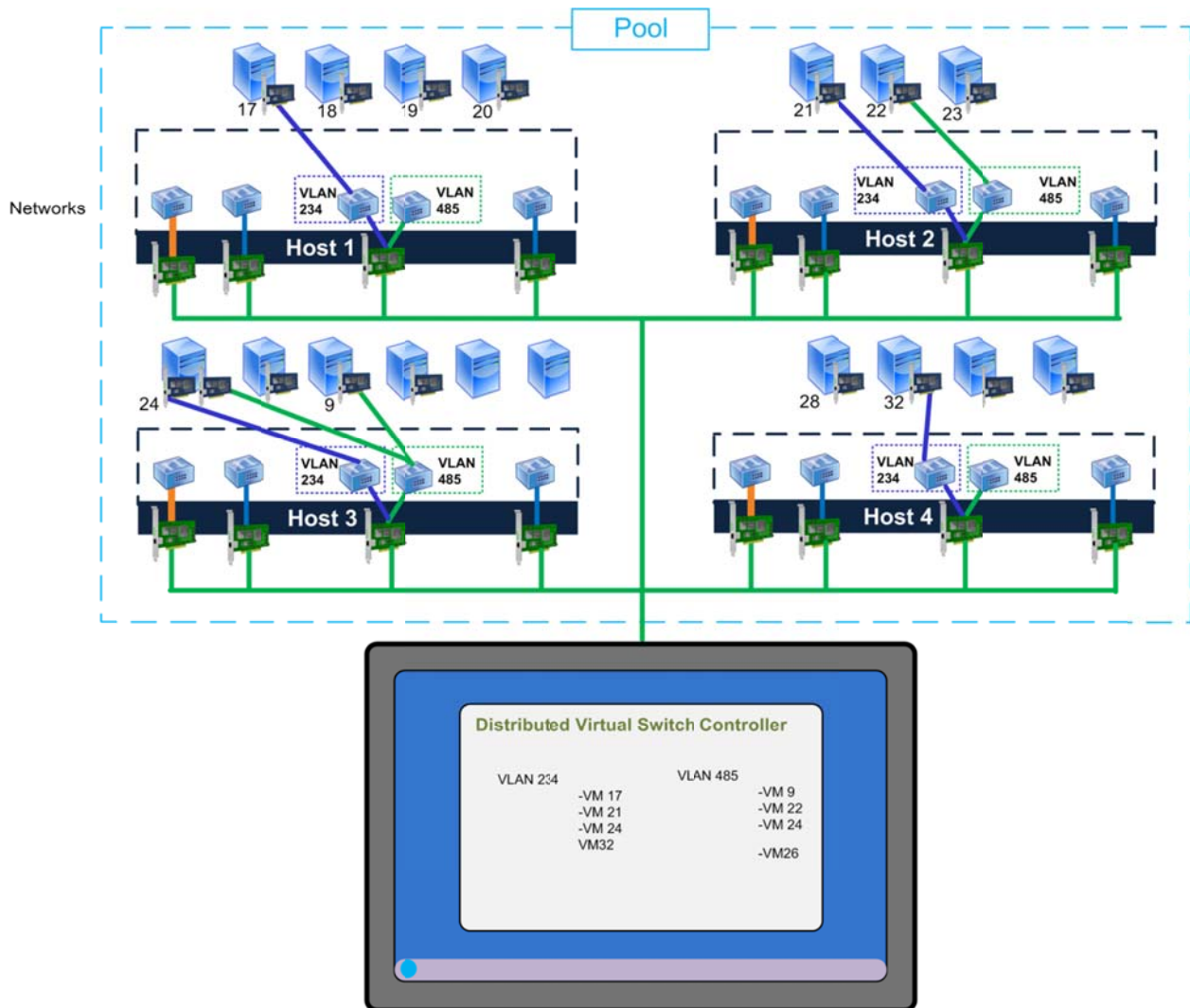
When designing your overall network configuration, you must determine your bonding and management-interface requirements; internal, external, and storage networking requirements; and VLAN configuration. You must also consider your IP addressing and DNS configuration. In addition, you should consider your NIC hardware configuration and other NIC settings, such as quality of service restrictions.

The sections that follow consider these decision points as they relate to the three primary design goals of redundancy, performance, and isolation.

Deciding to Use the Distributed Virtual Switch

As of XenServer 6.0, the new XenServer vSwitch component is the default networking configuration. However, you can still use the Linux bridge, which was the default networking configuration prior to XenServer 6.0, by running an XE command to change your networking configuration.

Citrix recommends that customers creating new XenServer pools use the XenServer vSwitch component for networking to prevent the need for future reconfiguration. As of XenServer 6.0, Citrix is discontinuing active development on the Linux bridge and may discontinue support for the Linux bridge in future releases.



This illustration shows how the vSwitch Controller can display which VLANs are used by which virtual machines and let you display these VLANs from one central user interface.



The Distributed Virtual Switch solution provides:

- Isolation through features such as Cross-Server Private Networks.
- Quality of Service (QoS) policies.
- Jumbo frame support.
- Fine-grained security policies. You can create access control lists by using the vSwitch Controller and restrict certain types of traffic to specific VMs.
- A central management console to manage finer grained features and monitor traffic. For the switch port associated with each VM, you can see the packets traversing that switch port.
- Visibility into XenServer networking through standard tools and protocols, such as RSPAN and NetFlow.
- Simplified administration of virtualized networking environments.

How does it work?

The Distributed Virtual Switch is a solution based on the Open vSwitch, an open-source project. The Distributed Virtual Switch comprises two networking components:

- **XenServer Open vSwitch.** The XenServer Open vSwitch, or “vSwitch,” is the actual networking component running on each XenServer host. The XenServer Open vSwitch is a virtualization-aware switch. This switch is referred to as an *Open vSwitch*.
- **Distributed vSwitch Controller.** The Distributed vSwitch Controller is console on a centralized server, which is distributed as an appliance, that manages and coordinates the behavior of each individual Open vSwitch to provide the appearance of a single distributed virtual switch. If you want to manage all the vSwitches on your hosts centrally and have them function as a single switch, you must download the Distributed vSwitch Controller appliance.

From a conceptual perspective, the vSwitch functions the same way as the existing Linux bridge. Regardless of whether or not you use the Distributed Virtual Switch or the Linux bridge, you can still use the same networking features in XenCenter and the same xe networking commands listed in the *XenServer Administrator's Guide*. In the diagrams throughout this guide, if you replace the existing “network” icons, which represent the Linux bridge, with a vSwitch, the concepts remain the same.

Because networking is a pool-level feature, if you want to use the Distributed Virtual Switch solution, you must configure all hosts in the pool to do so.

Configuring the Distributed Virtual Switch Solution

Citrix recommends setting up the networking configuration you want (vSwitch or Linux bridge) before you put your pool into production. The vSwitch is the default networking configuration, but



the Distributed Virtual Switch solution, which includes the vSwitch Controller Virtual Appliance, is not configured by default.

To ensure that XenServers can always reach an active vSwitch Controller, we recommend the use of Citrix High Availability for the vSwitch Controller VM. Refer to the XenServer Administrator's Guide for instructions on enabling high availability. Because continuous operation of the DVS Controller is critical to the operation of networking for all virtual machines, the vSwitch Controller VM restart-priority should be set to 1 and **ha-alwaysrun** should be set to **true**.

Configuring the Distributed Virtual Switch solution requires that you download the vSwitch Controller Virtual Appliance from [My Citrix.com](http://MyCitrix.com). The vSwitch Controller Virtual Appliance lets you manage your Distributed Virtual Switch implementation. For information about using the controller, see CTX130423 - [Citrix XenServer 6.0 vSwitch Controller User Guide](#).

Note: The Distributed Virtual Switch Controller is available in Citrix XenServer Advanced Edition or higher.

Reverting to the Linux Networking Configuration

If you want to revert to the legacy Linux networking bridge, run the following command on each host in the pool:

```
xe switch-network-backend bridge
```

Reboot the host after running this command.

Warning: The Linux network stack is not open flow enabled and does not support Cross Server Private Networks, and cannot be managed by the XenServer vSwitch Controller.

Important: Before switching networking configurations, shut down all the VMs on the host first.

Considerations:

- If you want to change the networking configuration, you must run the **xe-switch-network-backend** command on each host in the pool separately. The **xe-switch-network-backend** command is not a pool-wide command. This command can also be used to revert to the default vSwitch networking bridge by using this syntax: **xe-switch-network-backend openvswitch**.
- All hosts in the pool must use the same networking backend. Do not configure some hosts in the pool to use the Linux bridge and others to use the vSwitch bridge.
- When you are changing your hosts to use a different network configuration, you do not need to put the hosts into Maintenance mode. You just need to run the **xe-switch-network-backend** command on each host and reboot the hosts.

Note: After changing the networking configuration, check to make sure that your bonds are still enabled.



Designing Network Redundancy

As pressure on organizations to ensure network connectivity and availability for corporate resources increases, it is becoming important to improve network resiliency by ensuring redundancy at all network infrastructure levels. Citrix recommends that you consider the following network failure points when considering redundancy:

1. Network card.
2. Network cable (for example, if it is disconnected or damaged).
3. Switch (for example, if the power supply fails). You also might need to take switches offline for planned outages, such as firmware upgrades.

XenServer provides support for NIC bonding to provide network redundancy.

Considering NIC Bonding

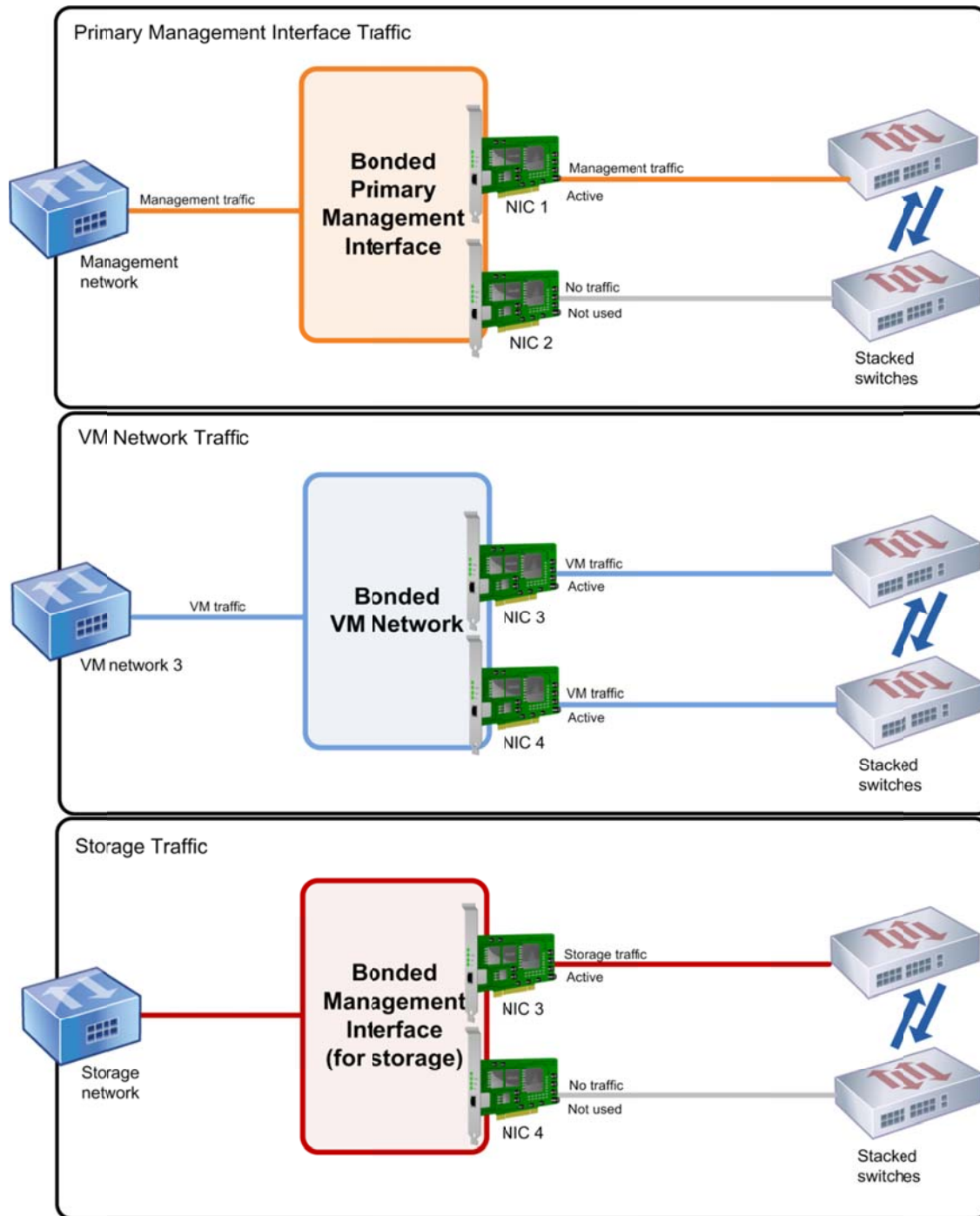
As previously discussed in “Creating Network Resiliency through Bonds,” XenServer lets you bond any combination of two NICs, including the ones that perform the following functions:

- **Primary management interfaces.** You can bond a primary management interface to another NIC so that the second NIC provides failover for management traffic. However, NIC bonding does not provide load balancing for management traffic.
- **NICs (non-management).** You can bond NICs that XenServer is using solely for VM traffic. Bonding these NICs not only provides resiliency, but doing so also balances the traffic from multiple VMs between the NICs.
- **Other management interfaces.** You can bond NICs that you have configured as management interfaces (for example, for storage). However, for most iSCSI software initiator storage, Citrix recommends configuring multipathing instead of NIC bonding since bonding management interfaces only provides failover without load balancing.

It should be noted that certain iSCSI storage arrays, such as Dell EqualLogic, require using bonds.

When considering whether or not to bond NICs, weigh your requirement for redundancy and load balancing against the number of separate subnets and VLANs each pool requires. Although you can configure XenServer with sixteen physical NICs per server bonding NICs reduces the number of physical networks you can connect into a host by half. XenServer supports a maximum of eight bonds.

The illustration that follows shows the differences between the three different types of interfaces that you can bond.



This illustration shows how, when configured in Active-active mode, the links that are active in bonds vary according to traffic type. In the top picture of a management network, NIC 1 is active and NIC 2 is passive. For the VM traffic, both NICs in the bond are active. For the storage traffic, only NIC 3 is active and NIC 4 is passive.



Selecting a Type of NIC Bonding

When you configure XenServer to route VM traffic over bonded NICs, by default, XenServer balances the load between the two NICs. However, XenServer does not require you to configure NIC bonds with load balancing (active-active). You can configure either:

- **Active-active bonding mode.** XenServer sends network traffic over both NICs in a load-balanced manner. Active-active bonding mode is the default bonding mode and without any additional configuration it is the one XenServer uses.
- **Active-passive bonding mode.** XenServer only sends traffic over one NIC in the bonded pair. If that NIC loses connectivity, the traffic fails over to the NIC that is not being used.

The best mode for your environment varies according to your environment's goals, budget, and switch performance. The sections for each mode discuss these considerations.

Note: Citrix strongly recommends bonding the primary management interface if the XenServer High Availability feature is enabled as well as configuring multipathing or NIC bonding for the heartbeat SR.

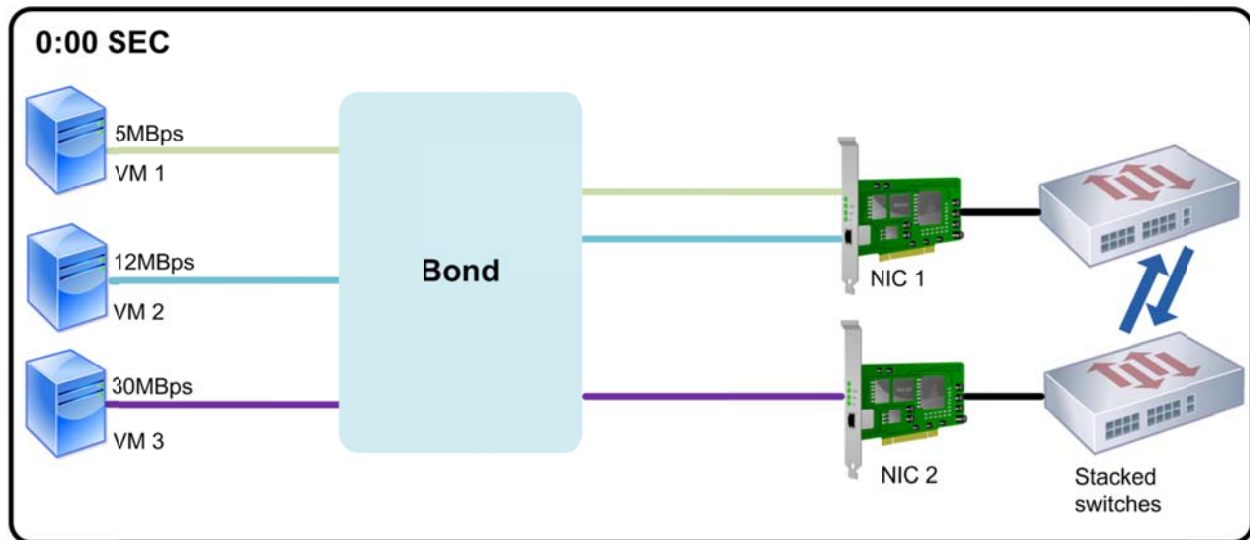
Understanding Active-Active NIC Bonding

When you bond NICs used for guest traffic in the default active-active mode, XenServer sends network traffic over both NICs in the bonded pair to ensure that it does not overload any one NIC with traffic.

XenServer does this by tracking the quantity of data sent from each VM's virtual interfaces and rebalancing the data streams every 10 seconds. For example, if three virtual interfaces (A, B, C) are sending traffic to one bond and one virtual interface (Virtual Interface B) sends more VM guest traffic than the other two, XenServer balances the load by sending traffic from Virtual Interface B to one NIC and sending traffic from the other two interfaces to the other NIC.

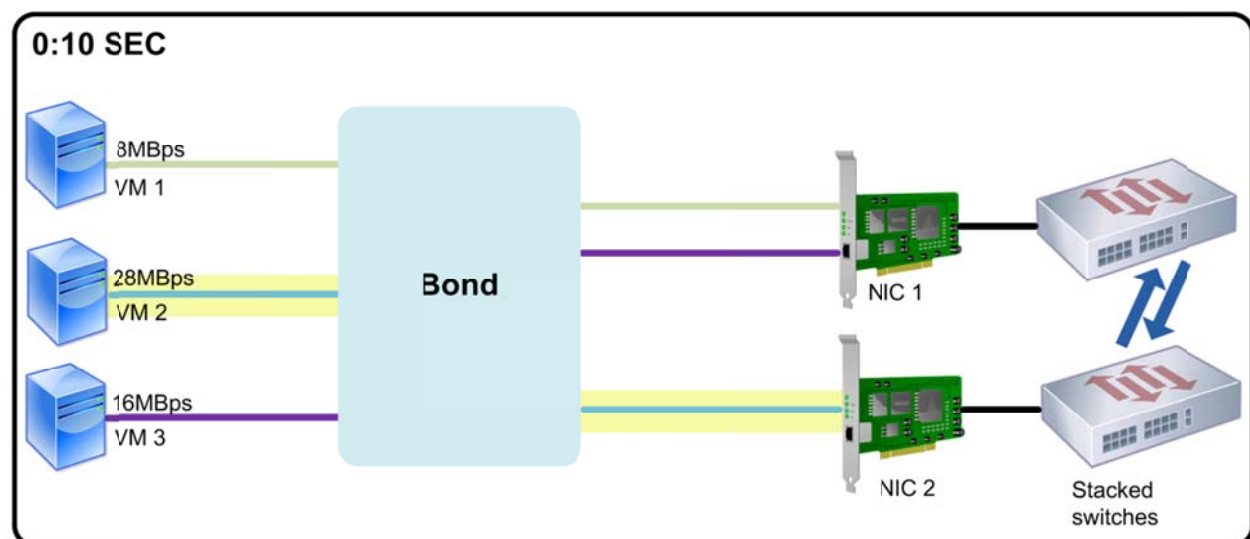
Important: When creating bonds, always wait until the bond is finished being created before performing any other tasks on the pool. To determine if XenServer has finished creating the bond, check the XenCenter logs.

The series of illustrations that follow show how XenServer redistributes VM traffic according to load every ten seconds.



In this illustration, VM 3 is sending the most data (30 megabytes per second) across the network, so XenServer sends its traffic across NIC 2. VM 1 and VM 2 have the lowest amounts of data, so XenServer sends their traffic over NIC 1.

The next illustration shows how XenServer reevaluates the load across the bonded pair after ten seconds.



This illustration shows how after ten seconds, XenServer reevaluates the amount of traffic the VMs are sending. When it discovers that VM 2 is now sending the most traffic, XenServer redirects VM 2's traffic to NIC 2 and sends VM 3's traffic across NIC 1 instead.

XenServer continues to evaluate traffic every ten seconds, so it is possible that the VM sending traffic across NIC 2 in the illustrations could change again at the twenty second interval.

Traffic from a single virtual interface is never split between two NICs.



The load balancing algorithm XenServer uses for active-active mode configurations is its own proprietary algorithm known as Source Level Balancing (SLB) NIC bonding. SLB is based on the open-source Linux Adaptive Load Balancing (ALB) mode.

Because SLB bonding is an active-active mode configuration, XenServer routes traffic over both NICs simultaneously. XenServer does not load balance management and IP-based storage traffic. For these traffic types, configuring NIC bonding only provides failover even when the bond is in active-active mode.

Note: XenServer NIC bonding does not require any switch configuration.

Understanding Active-Passive NIC Bonding

When XenServer is running NIC bonds in an active-passive configuration, XenServer routes traffic across one NIC in the bond only: this NIC is the only active NIC. XenServer does not send traffic over the other NIC in the bond so that NIC is *passive*, waiting for XenServer to redirect traffic to it if the active NIC fails.

To configure XenServer to route traffic on a bond in active-passive mode, you can use XenCenter and select **Active-passive** as the bond mode when you create the bond. You can also use the CLI to set a parameter on the master bond PIF (`xe bond-create mode=active-backup`), as described in the *XenServer Administrator's Guide*.

Note: `other-config:bond-mode=active-backup` is still supported for backwards compatibility.

When designing any network configuration, it is best to strive for simplicity by reducing components and features to the minimum required to meet your business goals. Based on this principle, consider configuring active-passive NIC bonding in situations such as the following:

- When you are connecting one NIC to a switch that does not work well with active-active bonding.

For example, if the switch does not work well with active-active bonding, you might see symptoms like packet loss, an incorrect ARP table on the switch, the switch would not update the ARP table correctly, and/or the switch would have incorrect settings on the ports (you might configure aggregation for the ports and it would not work).

- When you do not need load balancing or when you only intend to send traffic on one NIC. For example, if the redundant path uses a cheaper technology (for example, a lower-performing switch or external up-link) and that results in slower performance, configure active-passive bonding instead.

Note: As of XenServer 6.0, the vSwitch now supports active-passive NIC bonding. If you are using the vSwitch as your networking configuration, you can set the bonding mode to active-passive (also known as *active-backup*) using the XenCenter or the CLI.

Ensuring Resilience through Redundant Switches

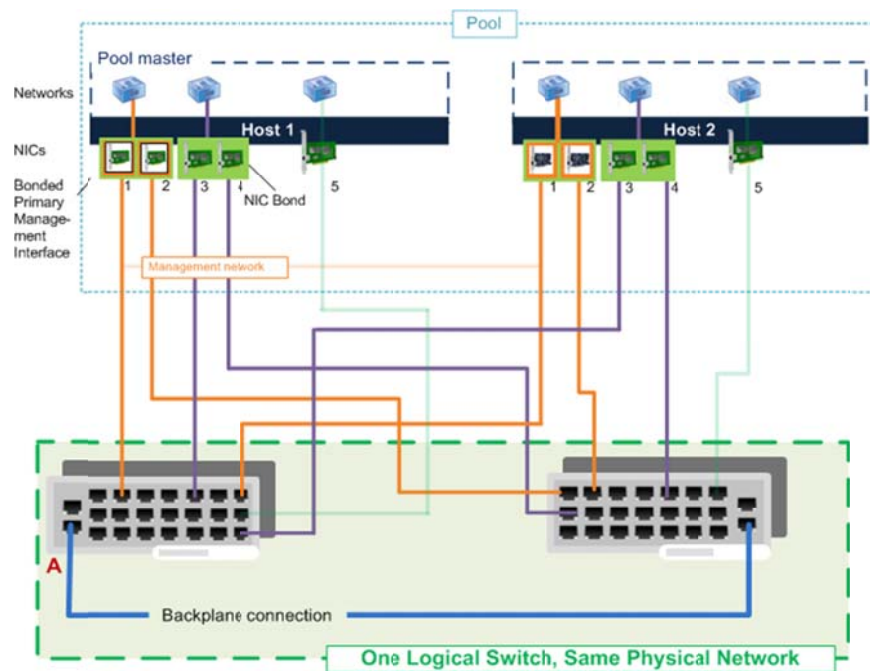
When you bond NICs, you can connect the links to either the same or separate switches, depending on your redundancy requirements. If you connect one of the links to a second, redundant switch and a NIC or switch fails, traffic fails over to the other link.

Adding a second switch helps in the following ways:

- When you bond NICs used exclusively for VM traffic, traffic is sent over both NICs. If you connect a link to a second switch and the NIC or switch fails, the virtual machines remain on the network since their traffic fails over to the other NIC/switch.
- When you connect one of the links in a bonded primary management interface to a second switch, it prevents a single point of failure for your pool. If the switch fails, the management network still remains online and the hosts can still communicate with each other.

When you attach bonded NICs to two switches, the switches must be running in a *stacked configuration*. (That is, the switches must be configured to function as a single switch that is seen as a single domain – for example, when multiple rack-mounted switches are connected across the backplane.) Switches must be in a stacked configuration because the MAC addresses of VMs will be changing between switches quite often while traffic is rebalanced across the two NICs.

The switches do not require any additional configuration. The illustration that follows shows how the cables and network configuration for the bonded NICs have to match.



This illustration shows how two NICs in a bonded pair use the same network settings, as represented by the networks in each host. The NICs in the bonds connect to different switches for redundancy.



Bonding Management Interfaces and MAC Addressing

Because bonds function as one logical unit, both NICs, regardless of whether the bond is active-active or active-passive, only have one MAC address between the two of them. That is, unless otherwise specified, the bonded pair uses the MAC address of the first NIC in the bond.

You can determine the first NIC in the bond as follows:

- In XenCenter, the first NIC in the bond is the NIC assigned the lowest number. For example, for a bonded NIC named “Bond 2+3,” the first NIC in the bond is NIC 2.
- When creating a bond using the `xe bond-create` command, the first PIF listed in the `piif-uuids` parameter is the first NIC in the bond.

When creating a bond, make sure that the IP address of the management interface before and after creating the bond is the same. When using DHCP, make sure that the MAC address of the management interface before creating the bond (that is, the address of one of the two NICs) is the same as the MAC of the bond after it is created.

From XenServer 6.0 onwards, if the MAC address is not specified (using the `mac` parameter when bonding from the CLI), the bond uses the MAC address of the primary management interface if the primary management interface is one of the interfaces in the bond. If another management interface (non-primary) is in the bond (and the the primary management interface is not), the bond uses the MAC address and IP address from that non-primary management interface. Otherwise, if none of the NICs in the bond is a management interface, the bond uses the MAC address of the first named NIC.

Note: After a pool is up and running, Citrix recommends using caution when bonding the primary management interface.

For more information about the `xe bond-create` command, see the *XenServer Administrator's Guide*.

Best Practices for Bonded Interfaces

Citrix strongly recommends configuring NIC bonding on the pool master after you join all member servers to the pool and before you create VMs. While it is technically possible to change NIC bonding afterwards, it can create issues.

Important: Do not join a host that already has a bond configured on it to a pool without first deleting the bond.

Ideally, for maximum performance, configure NIC bonds for VM traffic and isolate management traffic on its own network. However, note the following:



- Separating management, storage and VM traffic across different NICs helps prevent contention for network resources between the management, VM, and storage networks. However, unless you bond interfaces, they do not provide redundancy.
- Using the same bonded interface rather than two separate NICs for management and storage traffic can decrease performance, as all traffic will go through the same NIC.
- Always create bonds before creating virtual interfaces on VMs.

Warning: Do not attempt to bond NICs while the High Availability feature is enabled. Creating bonds can interrupt the in-progress High Availability heartbeat and cause hosts to self-fence (shut themselves down). Consequently, the hosts probably will not reboot correctly and will need the **host-emergency-ha-disable** command to recover.

Both NICs in the bond must have the same frame size (MTU). You can edit the frame size in XenCenter when you create the bond. For additional information, see “Configuring Networks with Jumbo Frames” on page 78.

Network card failures are rarely the reason network connections fail. Switch failures, network outages, and performance issues on one subnet are more common. Consider making bonded interfaces more reliable by:

- Configuring two different network up links (subnets) for each NIC in the bond. Not only is this a requirement, it helps ensure a network connection if there are issues (unrelated to hardware or switches) on one subnet.
- Connecting NICs bonded for management, storage, and guest traffic to different redundant switches.
- In an active-passive configuration, Citrix recommends connecting the passive NICs in each bond into one switch and the active NICs in each bond into a separate switch. If one of the switches fails, you still do not have a single point of failure because the failover NIC goes into another switch.

Where bonding requirements vary by workload type, consider grouping workloads with matching bonding requirements and NIC configurations in the same pool. Citrix makes this suggestion because XenServer automatically configures identical networking configurations across all hosts on the pool.

Designing Networks for Performance

If performance is a goal for your XenServer networking configuration, consider your I/O and performance requirements.

The I/O requirements of the workloads you are virtualizing determine how you should configure the physical NICs in each server (and, by extension, each pool).



Analyzing the traffic levels of the workloads may show, for example, that traffic levels let some hosts share a common external network while other workloads may require access to dedicated NICs to support their requirements. To improve throughput, you can implement NIC bonding: XenServer uses Source Load Balancing (SLB) to share load across bonded NICs.

It should be noted, however, that bonding has the *potential* to increase throughput only when VM traffic is not balanced between NICs. If VM traffic is already balanced between separate NICs, bonding will not increase throughput.

One of the most important factors for network performance is CPU utilization. The CPU utilization of your workloads has a significant impact on network throughput. As the CPU demands of workloads increase, the effective network performance may degrade. The impact of CPU utilization is discussed throughout this section.

Review the negative and beneficial impact sections that follow for ideas of ways you can optimize the performance of your XenServer network configuration.

Negative Impact

- **Many VLANs in a Pool.** If you have a lot of VLANs starting one or more hosts in a pool will be slower. Likewise, joining a host to a pool will be slower. However, having many VLANs will not affect network throughput.
- **Load on NICs.** Some models of network cards require firmware upgrades from the vendor to work reliably under load, or when certain optimizations are turned on. If you are seeing corrupted traffic to VMs, you should first try to obtain the latest recommended firmware from your vendor and apply a BIOS update. If this does not resolve your issue, contact Citrix Technical Support.

Potentially Beneficial

- **NIC Bonding.** By bonding two NICs together, you can increase the total amount of throughput available by better balancing the traffic. While bonding NICs does not affect the total bandwidth available -- the bandwidth available is the same as the combined bandwidth of the two individual NICs -- bonding can make better use of available resources.
- **Upgrading the NIC.** Provided infrastructure that can support such interfaces is in place, upgrading the NIC (for example, from a 1 gigabit NIC to a 10 gigabit NIC can improve performance).
- **Implementing jumbo frame** support can improve performance for storage traffic. For more information about using jumbo frames with storage, see “Chapter 6: Designing Your Storage Network Configuration” on page 70.

- **Implementing SR-IOV** for Provisioning Services server traffic. As of XenServer 5.6 Feature Pack 1, XenServer supports SR-IOV. For more information, see “Virtualizing the Provisioning Services Server” on page 81.

Testing XenServer Network Performance

You can observe a host’s network performance in several ways, including using the XenCenter **Performance** tab, Xen commands like **xentop** and **xenmon**, and networking testing tools, such as Iperf.

The XenCenter **Performance** tab displays the number of Mbps each NIC sends and receives and the load on the Control Domain, and the utilization for each CPU on the host. By clicking **Configure Graphs**, you can also change the Data Source to see network send and receive errors.

Before you begin testing the performance of your XenServer network configuration, you should understand what performance your hardware can theoretically support.

To test the performance of your XenServer network configuration on a specific XenServer host:

1. Start by using a network testing tool (such as Iperf).
2. Examine the throughput of your network, as your network testing tool recorded it. Compare the recorded throughput level with what your hardware can theoretically support. For example, if you know that your NIC supports 1 gigabit of throughput, then you should be getting a number close to that as your network throughput keeping in mind it is not possible to achieve the limit since there is always some overhead.
3. If you are not getting this level of throughput, do one or more of the following to find out more information:
 - In XenCenter, select the host, and use the **Performance** tab in XenCenter to check the following:

Make sure that VMs do not have a CPU, memory, or disk bottleneck because this prevents the VMs from receiving full throughput. The most common cause of reduced throughput is high CPU utilization by guests.
 - Run the **xentop** command. The **xentop** command shows you how much CPU each VM uses, network traffic to and from the VM, and disk traffic on the Control Domain. For more information about this command, enter **xentop -help** at the command prompt on your host.
 - Run the **xenmon** command. The **xenmon** command can help identify which domains and VMs are creating the highest I/O or processing loads on a host. For more information about this command, enter **xenmon.py -help** at the command prompt on your host.



What to do next?

1. Experiment with the number of processing threads (vs. the available virtual CPUs) on the VMs.
2. Reconsider your VM to host ratio. You may need to reduce the number of VMs on a host, or host a different mixture of workloads, to obtain the network performance you want.
3. Make sure there are no other VMs sending or receiving a lot of network traffic. Consider changing the home servers VMs are assigned to so they are balanced across different hosts according to their network utilization. You may also want to consider configuring Workload Balancing to make balancing recommendations or automatically balance VMs according to reads and writes.
4. Ensure VM traffic is evenly distributed across physical CPUs. For more information, see CTX127970 -- [Distributing Guest Traffic Over Physical CPUs in XenServer](#).
5. Load balance your network traffic across NICs. Make sure that no one NIC has an excessive number of VMs pointed to it and/or that these VMs are not sending an excessive amount of traffic.
6. Separate the management, storage, and VM traffic, as described in throughout this guide, to see if this improves performance.
7. If you have a mixed environment, consider putting a mixture of Linux and Windows VMs on that host. When virtualized, the Linux operating system usually puts less stress on the Control Domain and CPU resources.

Citrix recommends running these tests on a pilot environment before finalizing your hardware requirements and overall design. These tests may indicate that you need additional network connections, to rearrange workload groupings, to purchase more NICs, purchase NICs that support different configurations than you originally intended (for example, jumbo frames or SR-IOV) and so on.

Limiting Bandwidth Consumption for High Demand Workloads

In environments where the workloads send a lot of data and can potentially consume a lot of network bandwidth, you might want to limit the amount of data these workloads can send and slow down the transmission rate. This helps ensure other VMs on the host receive adequate network transfer rates.

To limit data transfer speeds, you can specify a maximum transfer rate in kilobytes per second (or kilobits per second in the CLI) when you create the VM's virtual interface(s) or at a later time. When you limit the transfer rate, you are doing so for all the data that VM sends over the virtual interface to its associated network link. If that VM uses multiple network links and you want to limit the transfer rate for all of them, you must create a QoS limit in each virtual interface for each network link.



Setting limits can be particularly useful when different organizations own different VMs on the same host since it helps ensure each VM gets a fair share of network bandwidth.

If you want to limit data transmission on more than one virtual interface on a VM, you must configure each virtual interface to have a QoS limit.

To configure a QoS limit for VM output, you have several options:

- **In XenCenter.** Select the VM, click the **Network** tab, and click either **Add Interface** or **Properties**.
- **In the vSwitch Controller.** For pools using the vSwitch and the Distributed Virtual Switching solution, configure the QoS setting in vSwitch Controller.
- **Using the vif-param-set xe command.** For example, to limit a virtual interface to a maximum transfer rate of 100 kilobits per second (kbps), use the command:

```
xe vif-param-set uuid=<vif_uuid> qos_algorithm_type=ratelimit
xe vif-param-set uuid=<vif_uuid> qos_algorithm_params:kbps=100
```

You do not need to create a QoS limit for all virtual interfaces/VMs that use that network link. You only need to set limits on the VMs you want to constrain.

Example: Calculating Bandwidth Limits

You can determine bandwidth limits using a formula like the following:

$$\text{Max Rate Transfer per VIF} = (\text{NIC speed in kilobits per second} / \text{number of VIFs}) * \text{desired network utilization \% (e.g., 70\%)}$$

An example of the application of this formula is as follows:

You have five VMs on a host with one virtual interface on each VM. The virtual interfaces all use the same 10 gigabit Ethernet NIC. You want to limit bandwidth consumption for all five VMs and their virtual interfaces.

Assuming a desired network utilization of 70%, you can theoretically limit each virtual interface bandwidth to 1,468,006 kilobits per second (or 183,500 kilobytes per second).

To obtain this number, we used the following calculations:

1. A 10Gigabit NIC = 10,485,760 kilobits per second
2. 10,485,760/5 virtual interfaces = 2,097,152 kilobits per second (per virtual interface)
3. 2,097,152 * 70% = 1,468,006 kilobits per second maximum transfer rate (per virtual interface)
4. 1,468,006 kilobits per second/8=183,500 kilobytes per second. XenCenter requires you enter the rate limit in kilobytes per second.



After determining the maximum transfer rate per virtual interface, you can then increase or decrease this value on each VM according to its bandwidth needs.

The best practice is not to guess at the network utilization, but rather measure the actual throughput from the NIC and divide it by the number of VMs using that NIC. If, for example, you are achieving 8,074,035 kilobits per second divide that by the number of VMs with a virtual interface configured to use that NIC. For example, $8,074,035/8$ VMs = an average throughput of 1,009,254 kilobits per second available for each VM.

You might decide to limit some VMs to far less than this value (for example, 800,000 kilobits per second) and others to more (for example, 1,200,000 kilobits per second) depending on the bandwidth requirements and business priority of the workloads.

However, be careful not to exceed the total amount of throughput achievable. If you do exceed it, the limits you set may not be reached.

In general, if you have not measured the throughput, it is better to set the maximum transfer rate slightly lower than what you expect your NICs can realistically achieve.

Note: The virtual interface QoS rate limit setting is different than the virtual disk QoS disk priority setting. The virtual disk QoS setting is a storage configuration that lets you assign priorities to virtual disks so that disks with higher priority are accessed faster than other disks. The *XenServer Administrator's Guide* describes both of these settings in more depth.

Additional Considerations

This section discusses an additional consideration for your networking configuration.

Enabling Promiscuous Mode for Traffic Monitoring

XenServer supports promiscuous mode, an Ethernet configuration for NICs in which the NIC receives all traffic on its link instead of only the frames addressed to that NIC's MAC address. Organizations may use promiscuous mode for a variety of reasons, such as requirements from transparent proxies or specialized traffic monitoring, security, and troubleshooting applications.

You can enable promiscuous mode at both the VM and XenServer host (physical server) levels. That is, you can configure promiscuous mode for both the virtual interfaces and physical NICs.

When you enable promiscuous mode on virtual interface or physical NIC, the mode lets you see all the traffic on a virtual switch. You might want to enable promiscuous mode when you want to:

- Run software (for example, software for a switch) or integrate an appliance that requires visibility into all traffic passing across the physical NIC to which it is connected. For configuration instructions, see CTX116493 -- [How to Enable Promiscuous Mode on a Physical Network Card](#).



- See all traffic going across the network (specifically, the virtual switch) between the NIC (PIF) and a VM's virtual interface. Instructions for implementing this configuration appear in CTX121729 -- [*How to Configure a Promiscuous Virtual Machine in XenServer*](#).

If your goal is to see all traffic VMs send across a specific network across a pool, you may need to configure promiscuous mode on a virtual interface on a VM in every host in the pool.

When determining if you want to enable promiscuous mode, consider the CPU load on the XenServer host. When XenServer runs in promiscuous mode, the kernel receives all network traffic and the CPU utilization on the host increases. Because this can slow down responses to network packets, this in turn can also increase network latency.

If you are using the vSwitch Controller introduced in XenServer 5.6 Feature Pack 1, you can use RSPAN instead of promiscuous mode to display traffic. The vSwitch Controller also includes functionality for mirroring.

Promiscuous Mode and Security

When promiscuous mode is configured on a virtual or physical switch port, the virtual interface or NIC connected to this switch port receives all traffic that travels through the switch and then passes it to the VM that “owns” the virtual interface (or, in the case of NICs, the Control Domain). This means that if malware or another type of malicious attack reaches across your network all VMs will become infected simultaneously.

Citrix recommends that you limit your use of promiscuous mode to troubleshooting and, possibly, security monitoring.

Warning: Do not enable promiscuous mode without a good reason since the VM in promiscuous mode can access the network traffic for other VMs.

Chapter 6: Designing Your Storage Network Configuration

This section is intended to highlight storage considerations you should include while designing your XenServer network configuration. It includes the following topics:

- The need to create a separate storage network
- How to assign IP addresses to NICs
- How to improve performance for storage traffic
- Guidance about choosing between NIC bonding and multipathing for storage

An extensive discussion of storage configurations is beyond the scope of this document.

Overview

This chapter is designed to help you configure your IP-based storage configuration so that it has redundancy and gets the best network performance with the lowest impact on other network traffic.

Specifically, this chapter explains how to create a separate storage network and when you would want to do so. It also explains the differences between the two failover choices for storage traffic, bonding and iSCSI multipathing. It also provides information about assigning IP addresses to NICs, since this is a common requirement for IP-based storage.

Other techniques discussed in this chapter that improve performance include jumbo frames.

Creating a Separate Storage Network

Citrix recommends dedicating one or more NICs as a separate storage network for NFS and iSCSI storage implementations. Many consider creating a separate storage network to be a best practice.

By configuring additional *management interfaces*, you can both assign an IP address to a NIC and isolate storage and network traffic, provided the appropriate physical configuration is in place. The term management interface refers to any NIC assigned an IP address for identification purposes. (The *primary* management interface, which is introduced in “Networking Configuration after Installation” on page 14, is a *type* of management interface.)

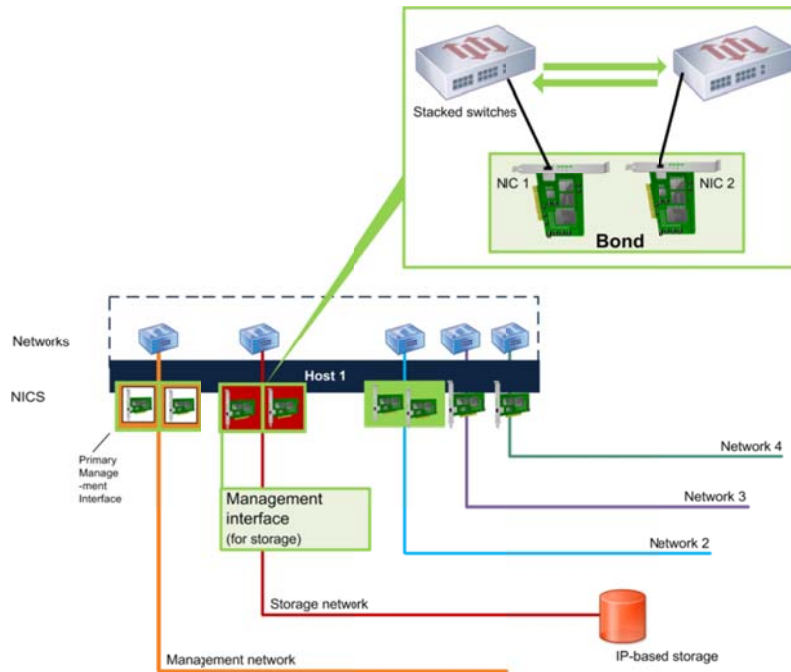
Tip: To figure out which NIC is the primary management interface, in XenCenter, click the **Network** tab, in the Management Interfaces section, check the Interfaces column for the word “Primary.”

You can segregate traffic by configuring an additional management interface(s) for storage and configure XenServer to access storage through that interface. Then, physically isolate the guest network and configure virtual machines to only use that isolated network. Ideally, the network should be bonded or use multipathing, as described in “Choosing to Enable Multipathing Support or Bond NICs.”

The overall process for creating a separate storage network is as follows:

1. Configuring physical network infrastructure so that different traffic is on different subnets.
3. Creating a management interface to use the new network.
3. Configuring redundancy, either multipathing or bonding.

In the illustration that follows, an administrator created a NIC bond from NICs 3 and 4 and then configured the bonded pair as a management interface for storage.



This illustration shows how the bond made from NICs 3 and 4 is configured as a management interface. XenServer sends its storage traffic over this NIC bond onto the storage network and, ultimately, the storage array. The exploded diagram shows how each NIC in the bond connects to a different switch.

Creating management interfaces lets you establish separate networks for, for example, IP-based traffic provided:

- You do not configure XenServer to use this network for any other purpose (for example, by pointing a virtual interface to this network).
- The appropriate physical network configuration is in place.

For example, to dedicate a NIC to storage traffic, the NIC, storage target, switch, and/or VLAN must be configured (physically connected) so that the target is only accessible over the assigned NIC.

To ensure that the storage traffic is separated from the management traffic, the storage network must be on a different subnet network. The subnet for storage must be a separate IP subnet that is not “routable” from the primary management interface. If the physical or logical configuration does not enforce the traffic separation, then XenServer may direct storage traffic over the primary management interface after a host reboot, due to the order in which XenServer initializes NICs.

In smaller environments, routing guest, management, and/or storage traffic together may not matter. However, in larger environments or environments with strict security policies, you may want to separate your traffic.

In order for IP-based storage, such as iSCSI software initiator and NFS, to communicate with XenServer, the XenServer NICs must have IP addresses. To specify an IP address for a NIC or



bond, you must create a management interface or reuse the IP address assigned to the primary management interface.

In other words, you can assign a XenServer IP address for your storage array to connect to by either:

- Configuring additional (non-primary) management interfaces so you can assign IP addresses to NICs besides the primary management interface for routing storage traffic.
- Routing storage traffic through the primary management interface -- since the primary management interface has an IP address this will work.

Some environments, such as test labs or at very small companies, may experience little impact from routing management and storage traffic on one interface. However, in general, Citrix strongly recommends that you do not route storage traffic over the primary management interface.

If you want to bond the management interface, create the management interface first and then bond it. Ideally, the first NIC in the bond should be the one for that has the IP address assigned to it. However, Citrix also recommends configuring the MAC address on the bond if you are using DHCP, as described in “Bonding Management Interfaces and MAC Addressing” on page 62.

Important: The primary management interface and other management interfaces must be on different subnets. This is especially critical when you are using the other management interfaces for storage traffic.

Assigning IP Addresses to NICs (Management Interfaces)

You can configure additional management interfaces in XenCenter and using the `xe` commands. To do so in XenCenter:

1. Ensure that the NIC is on a separate subnet, or routing is configured to suit your network topology in order to force the desired traffic over the selected NIC.
2. In the XenCenter resource pane, select the host that is the pool master, click the **Network** tab.

The screenshot shows the XenCenter interface with the Network tab selected. The 'Pool Networks' section contains a table with the following data:

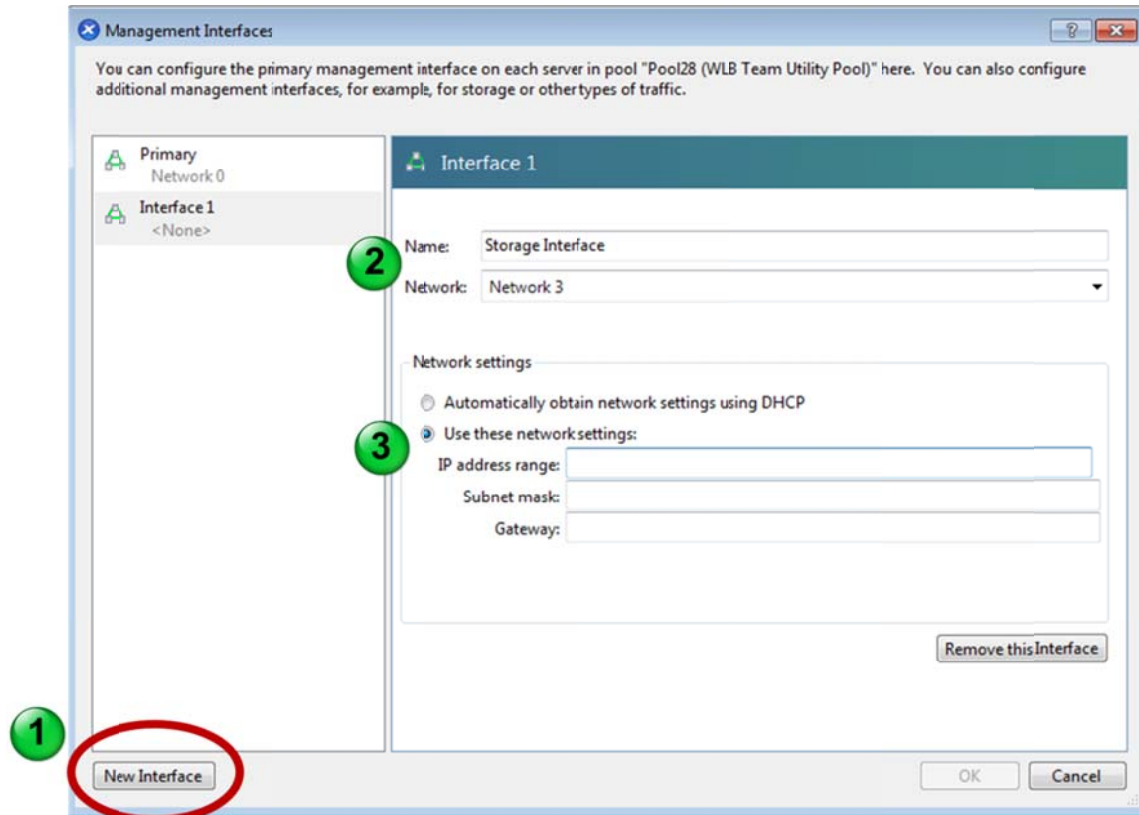
Name	Description	NIC	VLAN	Auto	Link Status	MAC	MTU
Network 0		NIC 0	-	Yes	Connected	00:19:b9:f2:96:00	1500
Network 1		NIC 1	-	Yes	Connected	00:19:b9:f2:96:75	1500
Test		NIC 2	1	No	Connected	-	1500
Network 2		NIC 2	-	Yes	Disconnected	00:15:17:48:a4:b0	1500
VLAN		NIC 2	6	No	Connected	-	1500

Below the network table is a 'Management Interfaces' section with the following table:

Server	Interface	Network	IP Address	Subnet mask	Gateway	DNS
host28	Primary	Network 0	10.204.154.28	255.255.255.0	10.204.154.1	10.9.3.22,10.204.6.51
host29	Primary	Network 0	10.204.154.29	255.255.255.0	10.204.154.1	10.9.3.22,10.204.6.51

A red circle highlights the 'Configure...' button located below the Management Interfaces table.

- The **Management Interfaces** feature is found on this tab. Instructions for configuring management interfaces varies by XenCenter release. See the *XenCenter Help* for more information.



Tip: If you want to dedicate a NIC for storage, check XenCenter to make sure that the NIC's associated network is not configured so that it is added to the VMs by default. To do so, in the **Networks** tab, right-click *<your-storage-network>* > **Properties**. Click **Network Settings** and make sure the **Automatically add this network to new virtual machines** check box is not selected.

Configuring Redundancy for Storage Traffic

For environments that want redundancy for their network storage traffic, XenServer supports NIC bonding and multipathing, including iSCSI HBA multipathing and software iSCSI multipathing. The term *multipathing* refers to routing storage traffic to a storage device over multiple paths for redundancy (failover) and increased throughput.

This section provides information about when to choose NIC bonding instead of multipathing and how to configure iSCSI multipathing.

Bonding NICs or configuring multipathing helps provide redundancy in case of partial network failure. Redundancy helps prevent interruptions to disk reads and writes. Interrupting disk reads and writes can lead to guest operating system failure if the VM consequently disconnects from the remote disk.

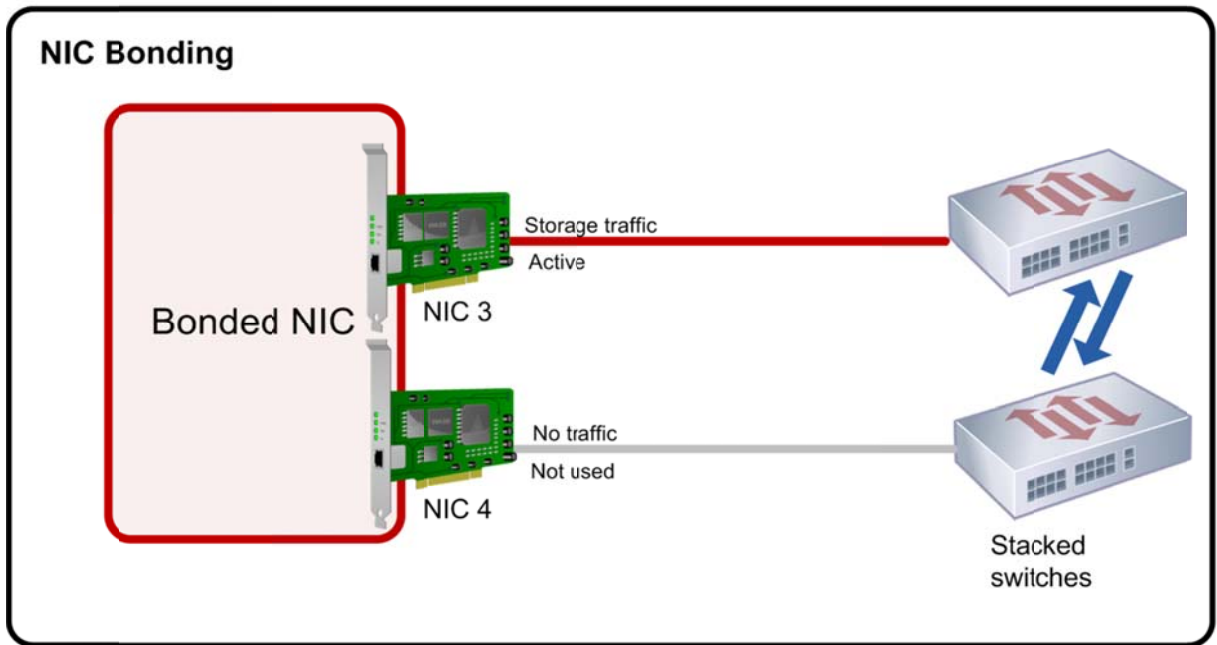
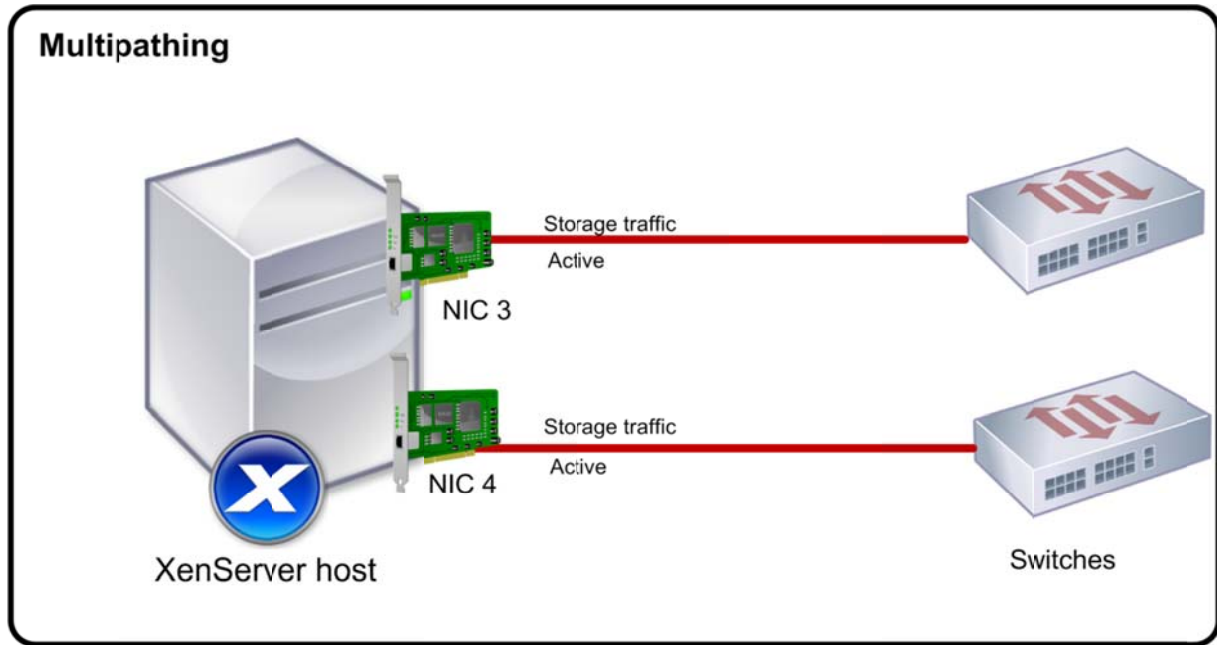
Citrix strongly recommends that you do not mix NIC bonding and iSCSI multipathing. There is no benefit from layering multipathing and NIC bonding on the same connection. After you enable multipathing, you not only have better performance but you also have the failover that bonding would have provided.

Note: XenServer supports multipathing for Fibre Channel and iSCSI SANs. However, multipathing for Fibre Channel SANs is not discussed in depth in this guide since traffic for Fibre Channel goes across a Fibre Channel network and not a TCP/IP network. For information about Fibre Channel multipathing, see the *XenServer Administrator's Guide* and the Citrix Knowledge Center.

Choosing to Enable Multipathing Support or Bond NICs

Citrix recommends configuring multipathing instead of NIC bonding whenever possible.

Like NIC bonding, multipathing provides failover for storage traffic. However, unlike NIC bonding, XenServer can send traffic down both paths when you configure multipathing: multipathing is an active-active configuration. By default, multipathing uses round-robin mode load balancing, so both routes will have active traffic on them during normal operation, which results in increased throughput. The illustration that follows provides a visual guide to the differences.



This illustration shows how, for storage traffic, both paths are active with multipathing whereas only one path is active with NIC bonding.

Citrix recommends using multipathing when you have block-based throughput (for example, iSCSI software initiator traffic). The exception to this is if, when the storage array is connected to XenServer, it does not work with multipathing.



Consider using NIC bonding instead of multipathing when:

- You have an NFS storage device.
- Your storage device does not support iSCSI connections over multiple IPs (for example, Dell EqualLogic or HP LeftHand SAN).

The following table shows the supported protocols for multipathing and NIC bonding:

	Multipathing	NIC Bonding
Supported Storage Protocols	Fibre Channel, iSCSI HBA, iSCSI software Initiator	NFS, CIFS, iSCSI software Initiator

Tip: To determine what XenServer hosts have multipathing enabled on them, check **Multipathing** in the **General** tab in XenCenter.

For information about configuring iSCSI multipathing, see CTX129309 -- [Configuring iSCSI Multipathing Support for XenServer](#).

Suggestions for Improving Storage Network Performance

This section provides guidance for improving storage network performance for iSCSI storage and jumbo frames.

Other methods of improving storage network performance were discussed in this chapter. These include creating a separate physical network for storage traffic, and when possible, choosing multipathing for redundancy since it both links are active, which improves throughput over NIC bonding.

iSCSI Storage

Provided iSCSI storage is configured correctly it can provide performance comparable to Fibre Channel.

iSCSI SANs consumes more CPU cycles on the host than NFS and Fibre Channel do. Because CPU utilization can affect network performance, this should be considered when sizing hardware and designing pools. However, it should also be noted that using an iSCSI host bus adaptor (HBA) can offload the processing for higher performance.

Citrix recommends using high performance network switches for iSCSI SANs to achieve better iSCSI performance. Citrix also recommends using redundant iSCSI network connectivity for all implementations.



It is particularly important to separate iSCSI storage traffic from management and VM guest traffic since it can interfere with non-storage traffic.

Configuring Networks with Jumbo Frames

Configuring XenServer networks to use jumbo frames can improve performance for storage traffic. Jumbo frames are Ethernet frames containing more than 1500 bytes of payload. Jumbo frames are typically used to achieve better throughput, reducing the load on system bus memory, and reducing the CPU overhead.

Currently, XenServer only supports jumbo frames if the vSwitch is used as the networking bridge on all hosts in the pool.

When determining whether your network performance will benefit from jumbo frames, consider the following:

- Jumbo frames can help offload CPU overhead.
 - By increasing the Ethernet frame size to 9000 bytes, jumbo frames reduce the number of packet headers the CPU must process, which consequently decreases the demands on the CPU. Jumbo frames also reduce the number of NIC interrupts needed when transmitting multi-packet file transfers.
 - Jumbo frames may help with slower CPUs if your NICs do not have a TCP Offload Engine (TOE) support. Jumbo frames are less apt to reduce CPU overhead with more intelligent gigabit NIC cards since these cards can process more of the packet headers on their own.
- Different types of workloads may have their own optimum packet sizes.
 - For example, for storage traffic, in general, throughput is more important than latency. As a result, if you configure larger packets for storage traffic (through jumbo frames), the storage traffic may not be negatively affected by latency and may benefit from the increased throughput.
 - For transactions that require high response times, enabling jumbo frames may not be helpful. Jumbo frames may require more buffering in the network, and, as a result, the latency for a particular item of data might be higher. (Larger packets take longer to assemble and more bandwidth per packet.) As a result, if you need high response times, bigger packets are not as good as smaller packets.

Generally, if the speed of the network and the need for high throughput increases, it is probably be good to increase the size of the packets.

- When the transfer size is, on average, relatively small, the benefits from jumbo frames might not be significant.



- The performance gains from jumbo frames vary according to the type of protocol and the type of traffic. When VM traffic is on the same network as storage traffic, if the VM traffic is latency sensitive, for example Voice-Over-IP traffic, configuring Quality of Service priorities in switches and on virtual interfaces may be necessary to ensure the performance of the voice traffic.
- Traffic that has relatively large block sizes at the application layer may benefit from jumbo frames since large block sizes make it easier for the TCP/IP stack to use large frames.

You can enable jumbo frame support on either of the following types of XenServer networks:

- External networks.
- Private networks. For example, you could enable jumbo frame support on a cross-server private network.

However, the equipment (NICs, switches) between all links must support jumbo frames. Certain types of traffic, such as IP-based storage traffic, may benefit from jumbo frame support. You can configure XenServer to use frames of between 1500 to 9216 Mbps.

When you create the network in which XenServer will transmit data in jumbo frames, you must specify the Maximum Transmission Unit (MTU) value the network supports. In XenCenter, you do so when you create the network in the **New Network** wizard by entering the value your entire network supports in the **MTU** text box.

Citrix strongly recommends specifying jumbo frames when you create the management interface for the storage network.

To implement jumbo frame support for bonded networks, you specify the MTU for both NICs in the bond when you create the bond. In XenCenter, there is an **MTU** text box in the **Create Bond** dialog box and in the **New Network** wizard.

Requirements for Using Jumbo Frames

- All NICs transmitting jumbo-frame traffic must support the transmission unit speed you want to configure and be on the *XenServer Hardware Compatibility List*.
- The network segment where you want to transmit the jumbo frames must have equipment (for example, the NICs and switches) that supports the frames in all segments, from end-to-end.
- When creating virtual interfaces and configuring NICs (PIFs), you must configure all interfaces on the network where you want to support jumbo frames with the same number of MTUs.

Citrix also recommends ensuring all components on the data path are tested for interoperability.



To achieve optimum performance, Citrix recommends that networks carrying jumbo frames use equipment with the same transmission speed (for example, 1 Gigabit). This is to promote the efficiency gains you achieve in Ethernet from standardization. While 10/100 Mbps networks may support jumbo frames, for optimum performance Citrix suggests using a minimum of 1 Gigabit Ethernet equipment. Ideally, networks should not use a mixture of 10/100 Mbps equipment and 1 Gigabit Ethernet equipment due to interoperability issues.

Additional Information

If you change MTU on the network in XenCenter, it will automatically change the MTU on the virtual interfaces and all the virtual interfaces and NICs/networks (switches) that point to the network supporting jumbo frames on all hosts.

Chapter 7: Considering Network Performance for PVS **– XenServer Deployments**

This chapter provides techniques for ensuring network performance when using XenServer to host Provisioning Services VMs. It includes the following topics:

- Network performance guidance when streaming Provisioning Services disk images to XenServer VMs
- Information about disabling the Spanning Tree protocol

This chapter assumes that you are familiar with how Provisioning Services basic works and its components. For an introduction to Provisioning Services, see the *Provisioning Services Installation and Configuration Guide* for your Provisioning Services version.

Virtualizing the Provisioning Services Server

Before deciding whether or not to host the Provisioning Services server on a VM, it is important to understand the bottlenecks and constraints of Provisioning Services servers, including how Provisioning Services servers use system cache. It is also important to understand that one of the primary resource bottlenecks of Provisioning Services servers is network I/O.

Recommendations

1. Citrix suggests that you host Provisioning Services servers on XenServer VMs that use Single Root I/O Virtualization (SR-IOV). SR-IOV lets a single PCIe device to appear as multiple separate physical PCIe devices. To use SR IOV, the NIC on the host must support SR-IOV. When configured, each VM behaves as though it is using the NIC directly, which reduces processing overhead.



Without configuring SR-IOV, it may be difficult to achieve throughput for Provisioning Services traffic over 2 gigabit speeds.

A Provisioning Services host could under-perform and result in a degraded user experience when there are high numbers of simultaneous VM startups.

To ensure solid network performance, it is critical to carefully consider the write cache configuration. In user workload scenario there are too many users working simultaneously, writing to and reading from the Provisioning Services write cache.

Configuring SR-IOV can mitigate this issue. For configuration information, see CTX126624 -- [*XenServer SR-IOV Support for Provisioning Services Virtual Machines*](#).

Note: SR-IOV is only supported on XenServer VMs that are running Windows Server 2008 as the guest operating system.

2. Ensure you are running at least XenServer 5.6 Feature Pack 1. This is to give more the Control Domain more CPU resources. As of XenServer 5.6 Feature Pack 1, the Control Domain can exploit multi-cores and is capable of using up to four cores.
3. Consider increasing the amount of memory allocated to the Control Domains on the hosts running the Provisioning Services VMs. You can increase the memory allocated to the Control Domain from the default allocation of 752MB by following CTX126531 -- [*Configuring Dom0 Memory in XenServer 5.6*](#).
4. If you want to virtualize Provisioning Services servers, consider reducing the target device to Provisioning Services server ratio so that the target devices do not exceed 300 targets. While virtualized Provisioning Services servers can support more hosts than this, reducing this ratio can improve performance.

See the Citrix Consulting white paper, CTX128645 -- [*Design Considerations for Virtualizing Citrix Provisioning Services*](#).

Note: It is not possible to perform live migration of Provisioning Services server VMs with SR-IOV configured. This is because the SR-IOV NIC's virtual function (VF) is directly tied to a specific virtual machine. (In XenServer, VFs are configured through the XenServer CLI whereas the VM template contains the configurations for the virtual interface functions and bridges in the Control Domain.) For High Availability design, you can use the Provisioning Services High Availability functionality instead of assuming live migration.

Additional Provisioning Services Networking Considerations

This section discusses some high-level IP addressing requirements and information about isolating the streaming service. There are additional considerations concerning the TCP Large Send Offload option. For more information about all of these topics, see CTX117374 -- [*Best Practices for Configuring Provisioning Server on a Network*](#).



IP Addressing Requirements

The Provisioning Server and Provisioning Services Target devices require at least two NICs with IP addresses on different networks:

- One network provides inter communication between devices specifically for the streaming I/O traffic.
- The other network provides access to network resources, the internet, and so on.

Isolating the Provisioning Services Streaming Service

Provisioning Services segment the stream traffic whenever applicable for several reasons: performance, provisioning growth and troubleshooting are a few.

When there are bottlenecks in network capacity or along the backplane of switches, UDP traffic tends to be the first to get dropped or discarded. As a result, Citrix encourages segmentation. When segmented, the streaming service does not have to compete for bandwidth, which ensures performance for the provisioned infrastructure. Segmentation can virtually eliminate retries and maximize Provisioning Services Target/Server performance.

Disabling the Spanning Tree Protocol

When deploying XenServer with Provisioning Services, as a best practice, disable the Spanning Tree Protocol on the switch (that is, set FastPort=On).

Citrix makes this recommendation because PXE takes initialize quickly and, consequently, means that it is best if the switch initializes as rapidly as possible to prevent the PXE environment from timing out. For similar reasons, Citrix recommends disabling the Spanning Tree protocol, since it is slow to initialize.

It is not necessary to hard code the default auto negotiation setting unless you notice long booting times and PXE timeouts.

Best Practice Configurations

Consider the following best practice guidelines when configuring XenServer –Provisioning Services deployments:

1. Follow the guidelines in CTX117374 -- [Best Practices for Configuring Provisioning Server on a Network](#).
2. Review the information about calculating IOPS for Provisioning Services servers in CTX125126 -- [Advanced Memory and Storage Considerations for Provisioning Services](#). **Note:** The guidance about CIFS no longer applies if you are running Windows Server 2008 R2.



3. If you want to deploy Provisioning Services on blades, review the Citrix Community Blog Post, "[Optimizing PVS.](#)" This article describes how to confine most Provisioning Services network traffic in the blade chassis so as to enable high-performance network connections between the blades.

Chapter 8: Verifying Your XenServer Networking Configuration

This chapter provides some basic steps for verifying that you configured and physically connected XenServer networking correctly. This chapter includes:

- Steps for checking your configuration
- An overview of steps you can take to resolve issues

Verifying XenServer Networking on a Pool

This chapter provides you with ways to verify your networking configuration before you create VMs and deploy the pool. Checking to see if your networking is configured properly before you proceed with further configuration can reduce troubleshooting and reconfiguration later.

Create a resource pool and run the tests in this chapter from any host in the pool. The tests in the first topic, “Verifying your Physical Configuration,” verify your physical configuration, including if your NICs are fully functioning. The tests in the other major topic, “Verifying your XenServer Networking Settings and Configuration,” verify whether your XenServer networking settings are configured correctly.

Verifying your Physical Configuration

This section provides procedures to help you verify NIC connectivity and NIC speed. For testing purposes only, create the following:

- A Windows VM and a Linux VM with XenServer Tools installed on them.
- An external network.



- An external bonded network.
- A XenServer single-server private network. Add two or more VMs (with XenServer Tools installed on them) to that network. You will need to manually configure IP addresses for these VMs unless a machine running DHCP was added to the private internal network.

Issues with NIC connectivity and NIC speed can indicate hardware that is not compatible with the [XenServer Hardware Compatibility List](#). They can also indicate issues with the NIC driver.

Verifying NIC Connectivity

1. Using Windows Remote Desktop Connection or a similar tool, verify that it is possible to connect to the Windows VM over both an external network and an external bonded network.
2. Using Iperf, verify connectivity between the VMs on the XenServer single-server private network.

Verifying NIC Speed

Testing to see if the NIC can send and receive traffic at the approximate speed for which it was rated can reveal issues, such as, for example, problems with the NIC drivers.

Before beginning the tests, download and install Iperf on a Windows VM, a Linux VM, and a separate physical server that is not running XenServer (a *control host*).

Run both the TCP bi-directional test and the UDP tests between each VM and the control host. For example, run the first test between (1) the Linux VM and the control host and (2) the Windows VM and the control host.

TCP Bi-Directional Test

1. On the Windows VM, run the following command with the duration set to 300 seconds:

```
iperf.exe -c 192.168.1.1 -d -t 300
```
2. While running the command on the VM, run the following command on the control host:

```
iperf.exe -s
```
3. Repeat steps 1 and 2 using the Linux VM instead of the Windows VM.
4. Repeat steps 1 to 3 using a NIC bond.



UDP Test

1. On the Linux VM, run the following command with the duration set to 300 seconds. In this test, the VM functions as the Iperf client.

```
iperf.exe -c 192.168.1.1 -u -t 300 -b <bandwidth>
```

Replace <bandwidth> with “100M” for a 1 gigabit NIC. For a 10 gigabit NIC, replace <bandwidth> with “1000M”.

2. While running the command on the VM, run the following command on the control host:

```
iperf.exe -s -u
```

3. Repeat steps 1 and 2 using the Windows VM instead of the Linux VM.
4. Repeat steps 1 to 3 except using each VM as the Iperf server and the control host as the Iperf client.
5. Repeat steps 1 to 4 using a NIC bond.

Verifying your XenServer Networking Settings and Configuration

After setting up your XenServer networking configuration, verify that you did it correctly by checking the following:

1. Management Interfaces.

Configure networking after forming the pool, as described in “Sequence of Networking Configuration Tasks” on page 18. After creating the initial networking configuration or adding hosts to a pool, it is possible for the primary management interfaces on the other hosts in the pool to be incompletely configured (for example, if the pool master does not successfully propagate its settings to the member or joining servers).

1. Verify that each primary management interface uses the same (corresponding) NIC on each XenServer host.
 2. Verify that you set a unique IP address on each XenServer primary management interface or bonded pair of primary management interfaces.
 3. Verify a) and b) on any additional management interfaces you configured (for example, for storage)
2. Verify all the NICs you bonded were created correctly. To do so, check the XenCenter logs for failures or any bonds that unfinished. A typical sign that bonding did not work is that the bond takes excessively long to finish or it seems to “hang.”

3. Verify that the networking configuration and physical cabling on each XenServer host matches. For example, is NIC 3 on Host 4 configured with the same networks and cabling as NIC 3 on Host 5.

Tip: To make an LED light for a NIC on a XenServer host blink, run the **ethtool** command with the **-p** option on the host. **ethtool** is a Linux command that displays or changes NIC settings. For example, to use the **ethtool** command to make the lights on the first NIC on the host, run the following on the host: `ethtool -p eth0 10`. In this example, **eth0** is the NIC and **10** is the number of seconds you want the NIC to blink.

4. In XenCenter, select each host individually, and verify the information in the **NICs** and **Network** tabs is fully present.

In the **NICs** tab, there should be a row in the NICs list for each NIC. Verify the columns in the list have correct values, especially the speed and MAC address columns.

In the **Networks** tabs, verify that all the NICs listed in the **NICs** tab are present and they have the correct network listed beside them.

5. Verify that the physical networking configuration matches for each XenServer host (that is, check to make sure that NIC 3 on Host 4 is cabled to have the same connectivity as NIC 3 on Host 5 and so on).
6. If you are using a vSwitch as the networking bridge, verify that it is enabled on all hosts in the pool.
7. Install two VMs on different hosts, if necessary, and put them on the same network to see if they can “ping” each other.

Resolving Issues

If, after checking these items, VMs or hosts cannot connect to each other as expected, then:

1. Verify that your physical configuration (cables and switches) is correct.

Important: The #1 networking issue the XenServer technical support team sees is not actually a *XenServer* issue. Most calls are from customers with incorrectly configured physical networks (cabling errors) or incorrectly configured switches. Before calling Citrix Technical Support, verify your cabling configuration.

1. Review the section, “Recovering from a bad network configuration” in the *XenServer Administrator’s Guide*.

Important: After the pool is in production, always use extreme care when changing the physical networking hardware in a host. Before changing, adding, or replacing any NICs on the host,



always take the VMs offline and put the host into maintenance mode so that the host is no longer connected to any storage repositories.



Revision History

Revision	Comments	Date
1.0	Initial Release	April 28, 2011
2.0	Updated for XenServer 6.0 release. Also, added appendix about changing your networking configuration.	September 23, 2011



About Citrix

Citrix Systems, Inc. (NASDAQ:CTXS) is the leading provider of virtualization, networking and software as a service technologies for more than 230,000 organizations worldwide. Its Citrix Delivery Center, Citrix Cloud Center (C3) and Citrix Online Services product families radically simplify computing for millions of users, delivering applications as an on-demand service to any user, in any location on any device. Citrix customers include the world's largest Internet companies, 99 percent of Fortune Global 500 enterprises, and hundreds of thousands of small businesses and prosumers worldwide. Citrix partners with over 10,000 companies worldwide in more than 100 countries. Founded in 1989, annual revenue in 2010 was \$1.87 billion.

©2011 Citrix Systems, Inc. All rights reserved. Citrix®, Access Gateway™, Branch Repeater™, Citrix Repeater™, HDX™, XenServer™, XenCenter™, XenApp™, XenDesktop™ and Citrix Delivery Center™ are trademarks of Citrix Systems, Inc. and/or one or more of its subsidiaries, and may be registered in the United States Patent and Trademark Office and in other countries. All other trademarks and registered trademarks are property of their respective owners.