

INFORMATION GUIDE

VMware Virtual Networking Concepts



Table of Contents

Introduction..... 3

ESX Server Networking Components..... 3

How Virtual Ethernet Adapters Work..... 4

How Virtual Switches Work..... 4

A Virtual Switch is Similar to a Physical Switch 4

A Virtual Switch is Different from a Physical Switch..... 5

Spanning Tree Protocol Not Needed 5

Virtual Switch Isolation 5

Virtual Ports 5

Uplink Ports 6

Port Groups..... 6

Uplinks 6

Virtual Switch Correctness 7

VLANs in VMware Infrastructure 7

NIC Teaming 8

Load Balancing 8

Failover Configurations 9

Layer 2 Security Features 10

Managing the Virtual Network..... 10

Appendix: Virtual Device Maximums 11

VMware Virtual Networking Concepts

Introduction

VMware Infrastructure 3 provides a rich set of networking capabilities that integrate well with sophisticated enterprise networks. These networking capabilities are provided by VMware ESX Server and managed by VMware VirtualCenter.

With virtual networking, you can network virtual machines in the same way that you do physical machines and can build complex networks within a single ESX Server host or across multiple ESX Server hosts, for production deployments or development and testing purposes.

Virtual switches allow virtual machines on the same ESX Server host to communicate with each other using the same protocols that would be used over physical switches, without the need for additional networking hardware. ESX Server virtual switches also support VLANs that are compatible with standard VLAN implementations from other vendors.

A virtual machine can be configured with one or more virtual Ethernet adapters, each of which has its own IP address and MAC address. As a result, virtual machines have the same properties as physical machines from a networking standpoint.

In addition, virtual networks enable functionality not possible with physical networks today.

This guide is for VMware Infrastructure 3 users who want a clearer understanding of the basic design of the networking capabilities in VMware Infrastructure 3 and how that design affects deployment in the datacenter.

VMware Infrastructure Networking Components

The VMware Infrastructure 3 networking stack uses a modular design for maximum flexibility.

The key virtual networking components provided by Virtual Infrastructure 3 are virtual Ethernet adapters, used by individual virtual machines, and virtual switches, which connect virtual machines to each other and connect both virtual machines and the ESX Server service console to external networks, as shown in Figure 1.

The sections that follow provide more detail about each of these components.

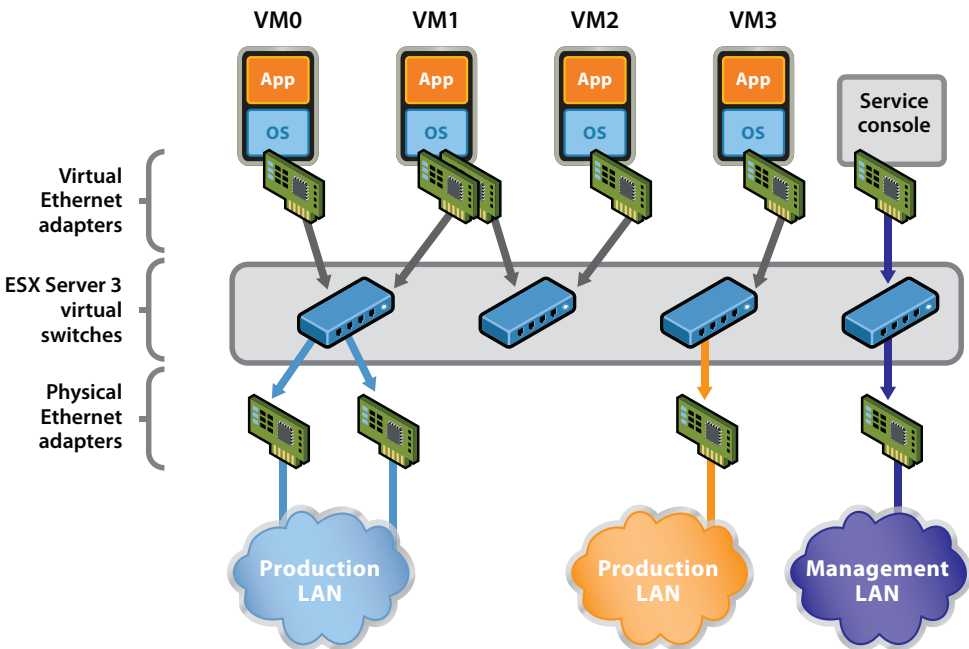


Figure 1 — Virtual switches in ESX Server 3 connect virtual machines and the service console to each other and to external networks.

How Virtual Ethernet Adapters Work

In discussions of VMware Infrastructure 3, you may see references to as many as five different virtual network adapters. Three of those are virtual Ethernet adapters used by virtual machines. In most cases, a virtual machine uses only one of the three types of virtual adapters. The three types of adapters available for virtual machines are:

- **vmxnet** — a paravirtualized device that works only if VMware Tools is installed in the guest operating system. A paravirtualized device is one designed with specific awareness that it is running in a virtualized environment. The vmxnet adapter is designed for high performance. In most cases, when you select the flexible network adapter, this is the adapter used after VMware Tools is installed in the guest operating system.
- **vlance** — a virtual device that provides strict emulation of the AMD Lance PCNet32 Ethernet adapter. It is compatible with most 32-bit guest operating systems. This adapter is used when you select the flexible network adapter but VMware Tools is not installed in the guest operating system.
- **e1000** — a virtual device that provides strict emulation of the Intel E1000 Ethernet adapter. This is the virtual Ethernet adapter used in 64-bit virtual machines. It is also available in 32-bit virtual machines.

The other virtual network adapters are:

- **vswif** — a paravirtualized device similar to vmxnet that is used only by the ESX Server service console.
- **vmknic** — a virtual device in the VMkernel, the software layer that manages most of the physical resources on the ESX Server host. The vmknic is used by the TCP/IP stack that services VMotion, NFS and software iSCSI clients that run at the VMkernel level, and remote console traffic.

All five of the virtual network devices share the following characteristics:

- They have their own MAC addresses and unicast/multicast/broadcast filters.
- They are strictly Layer 2 Ethernet adapter devices.

Note: The speed and duplex settings found in physical networking are not relevant in the virtual network, because all the data transfer takes place in the host system's RAM, nearly instantaneously and without the possibility of collisions or other signaling-related errors.

How Virtual Switches Work

Virtual switches are the key networking components in VMware Infrastructure 3. You can create up to 248 virtual switches on each ESX Server 3 host.

A virtual switch is "built to order" at run time from a collection of small functional units. Some of the key functional units are:

- The core Layer 2 forwarding engine.
 - This is a key part of the system (for both performance and correctness), and in Virtual Infrastructure 3 it is simplified so it only processes Layer 2 Ethernet headers. It is completely independent of other implementation details, such as differences in physical Ethernet adapters and emulation differences in virtual Ethernet adapters.
- VLAN tagging, stripping, and filtering units.
- Layer 2 security, checksum, and segmentation offload units.

This modular approach has become a basic principle to be followed in future development, as well.

When the virtual switch is built at run time, ESX Server 3 loads only those components it needs. It installs and runs only what is actually needed to support the specific physical and virtual Ethernet adapter types used in the configuration. This means the system pays the lowest possible cost in complexity and demands on system performance.

The design of ESX Server 3 supports temporarily loading certain components in the field — a capability that could be used, for example, for running appropriately designed diagnostic utilities.

An additional benefit of the modular design is that VMware and third-party developers can easily incorporate modules to enhance the system in the future.

In many ways, the ESX Server virtual switches are similar to physical switches. In some notable ways, they are different. Understanding these similarities and differences will help you plan the configuration of your virtual network and its connections to your physical network.

A Virtual Switch Is Similar to a Physical Switch

A virtual switch, as implemented in ESX Server 3, works in much the same way as a modern Ethernet switch.

It maintains a MAC:port forwarding table and performs the following functions:

- Looks up each frame's destination MAC when it arrives.
- Forwards a frame to one or more ports for transmission.
- Avoids unnecessary deliveries (in other words, it is not a hub).

An ESX Server 3 virtual switch supports VLAN segmentation at the port level. This means each port can be configured in either of the following ways:

- With access to a single VLAN, making it what's called an access port in the world of physical switches, or in ESX Server terminology, using virtual switch tagging.
- With access to multiple VLANs, leaving tags intact, making it what's called a trunk port in the world of physical switches, or in ESX Server terminology, using virtual guest tagging.

For more information on these options, see the section VLANs in VMware Infrastructure on page 7.

An ESX Server 3 virtual switch supports copying packets to a mirror port. By using what is called promiscuous mode, ESX Server makes a virtual switch port act as a SPAN port or mirror port. This capability makes it possible to debug using a sniffer or to run monitoring applications such as IDS.

In addition, an administrator can manage many configuration options for the switch as a whole and for individual ports using the Virtual Infrastructure Client.

A Virtual Switch Is Different from a Physical Switch

ESX Server provides a direct channel from virtual Ethernet adapters for such configuration information as authoritative MAC filter updates. So there is no need to learn unicast addresses or perform IGMP snooping to learn multicast group membership.

Ports on the virtual switch may automatically enter mirror mode when the virtual Ethernet adapter's promiscuous bit is set — if virtual switch and port group policies allow.

Spanning Tree Protocol Not Needed

VMware Infrastructure 3 enforces a single-tier networking topology. In other words, there is no way to interconnect multiple virtual switches, thus the network cannot be configured to introduce loops. As a result, Spanning Tree Protocol (STP) is not needed and is not present.

Note: It is actually possible, with some effort, to introduce a loop with virtual switches. However, to do so, you must run Layer 2 bridging software in a guest with two virtual Ethernet adapters connected to the same subnet. This would be difficult to do accidentally, and there is no reason to do so in typical configurations.

Virtual Switch Isolation

Network traffic cannot flow directly from one virtual switch to another virtual switch within the same host. Virtual switches provide all the ports you need in one switch, leading to the following benefits:

- Because there is no need to cascade virtual switches, Virtual Infrastructure 3 provides no capability to connect virtual switches.
- Because there is no way to connect virtual switches, there is no need to prevent bad virtual switch connections.
- Because virtual switches cannot share physical Ethernet adapters, there is no way to fool the Ethernet adapter into doing loopback or some similar configuration that would cause a leak between virtual switches.

In addition, each virtual switch has its own forwarding table, and there is no mechanism to allow an entry in one table to point to a port on another virtual switch. In other words, every destination the switch looks up can match only ports on the same virtual switch as the port where the frame originated, even if other virtual switches' lookup tables contain entries for that address.

It is unlikely that a would-be attacker could circumvent virtual switch isolation because it would be possible only if there were a substantial unknown security flaw in the vmkernel. Because ESX Server parses so little of the frame data — primarily just the Ethernet header — this would be difficult, and once an attacker had such access, richer targets than breaking virtual switch isolation are readily available.

There are natural limits to this isolation. If you connect the uplinks of two virtual switches together, or if you bridge two virtual switches with software running in a virtual machine, you open the door to the same kinds of problems you might see in physical switches.

Virtual Ports

The ports on a virtual switch provide logical connection points among virtual devices and between virtual and physical devices. You can think of them as virtual RJ-45 connectors. Each virtual switch can have up to 1,016 virtual ports, with a limit of 4,096 ports on all virtual switches on a host.

The virtual ports in ESX Server provide a rich control channel for communication with the virtual Ethernet adapters attached to them. ESX Server virtual ports:

- Know authoritatively what the configured receive filters are for virtual Ethernet adapters attached to them. This means no MAC learning is required to populate forwarding tables.
- Unlike physical switches, know authoritatively the "hard" configuration of the virtual Ethernet adapters attached to them. This capability makes it possible to set such policies as "guest can't change MAC address," because the virtual switch port can essentially know for sure what is "burned into ROM" (actually, stored in the configuration file, outside control of the guest operating system).

Uplink Ports

Uplink ports are ports associated with physical adapters, providing a connection between a virtual network and a physical network. Physical adapters connect to uplink ports when they are initialized by a device driver or when the teaming policies for virtual switches are reconfigured.

Some virtual switches should not connect to a physical network and thus have no uplink port, as shown in Figure 2. This is the case, for example, for a virtual switch that provides connections between a firewall virtual machine and the virtual machines protected by the firewall.

Virtual Ethernet adapters connect to virtual ports when you power on the virtual machine on which the adapters are configured, when you take an explicit action to connect the device, or when you migrate a virtual machine using VMotion.

A virtual Ethernet adapter updates the virtual switch port with MAC filtering information when it is initialized and whenever it changes.

A virtual port may ignore any requests from the virtual Ethernet adapter that would violate the Layer 2 security policy in effect for the port. For example, if MAC spoofing is blocked, the port drops any packets that violate this rule.

Port Groups

As important as they are in VMware Infrastructure virtual networking, port groups do not correspond exactly to features commonly found in physical networks. The closest counterpart is the SmartPort feature offered in some Cisco switches. You can think of port groups as templates for creating virtual ports with particular sets of specifications. You can create a maximum of 512 port groups on a single host.

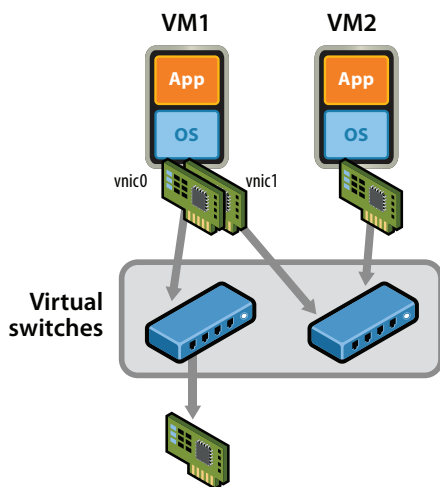


Figure 2 — Virtual switches in ESX Server 3 can be configured with uplink ports or with no uplink port.

Port groups are important particularly for VMotion. To understand why, consider what happens as virtual machines migrate to new hosts using VMotion.

Port groups make it possible to specify that a given virtual machine should have a particular type of connectivity on every host on which it might run.

Port groups are user-named objects that contain enough configuration information to provide persistent and consistent network access for virtual Ethernet adapters:

- Virtual switch name
- VLAN IDs and policies for tagging and filtering
- Teaming policy
- Layer 2 security options
- Traffic shaping parameters

In short, port group definitions capture all the settings for a switch port. Then, when you want to connect a virtual machine to a particular kind of port, you simply specify the name of a port group with an appropriate definition.

Port groups may specify different host-level parameters on different hosts — teaming configurations, for example. But the key element is that the result is a consistent view of the network for a virtual machine connected to that port group, whichever host is running it.

Note: Port groups do not necessarily correspond one-to-one to VLAN groups. It is possible, and even reasonable, to assign the same VLAN ID to multiple port groups. This would be useful if, for example, you wanted to give different groups of virtual machines different physical Ethernet adapters in a NIC team for active use and for standby use, while all the adapters are on the same VLAN.

Uplinks

Physical Ethernet adapters serve as bridges between virtual and physical networks. In VMware Infrastructure, they are called uplinks, and the virtual ports connected to them are called uplink ports. A single host may have a maximum of 32 uplinks, which may be on one switch or distributed among a number of switches.

In order for a virtual switch to provide access to more than one VLAN, the physical switch ports to which its uplinks are connected must be in trunking mode. It is important to prune the VLANs, keeping only those that are required for the virtual switch. Failure to do so can cause unnecessary overhead on the ESX Server host because it must process broadcast traffic for all VLANs trunked to it.

You should prune VLANs at the physical switch level, but pruning at the physical switch cannot be quite as aggressive as pruning at the uplink because the virtual switch knows which

virtual machines are actually powered on. As a result, the virtual switch may be able to prune VLANs that are needed but are not in use at the time you are pruning.

You can specify different teaming behavior for different groups of virtual machines that share the same team of physical adapters. For example, you can vary the active/standby status of each adapter in the team across port groups to get both good link aggregation and failover behavior. For more information on teaming, see NIC Teaming on page 8.

Teaming state — which physical Ethernet adapters are actually transporting data — is maintained for each port group. Teaming state transitions are mostly transparent to virtual Ethernet adapters. Virtual machines cannot tell when a failover has occurred or which physical adapter is carrying any given frame. When the transition removes or restores actual access to a physical network — that is, when the last link goes down or the first link comes up — the network visibility change is apparent to guests.

Uplinks are not required for a virtual switch to forward traffic locally. Virtual Ethernet adapters on the same virtual switch can communicate with each other even if no uplinks are present. If uplinks are present, they are not used for local communications within a virtual switch.

When VLANs are configured, ports must be on the same VLAN in order to communicate with each other. The virtual switch does not allow traffic to pass from one VLAN to another. Communication between VLANs is treated the same as communication between virtual switches — it is not allowed. If you do want communication between two VLANs or two virtual switches, you must configure an external bridge or router to forward the frames.

Virtual Switch Correctness

Two correctness issues are particularly important. It is important to ensure that virtual machines or other nodes on the network cannot affect the behavior of the virtual switch. ESX Server guards against such influences in the following ways:

- Virtual switches do not learn from the network in order to populate their forwarding tables. This eliminates a likely vector for denial of service or leakage attacks, either as a direct denial of service attempt or, more likely, as a side effect of some other attack, such as a worm or virus, as it scans for vulnerable hosts to infect.
- Virtual switches make private copies of any frame data used to make forwarding or filtering decisions. This is a critical feature of the virtual switch and is unique to virtual switches.

The virtual switch does not copy the entire frame, because that would be inefficient, but ESX Server must make sure that the guest operating system does not have access to any sensitive data once the frame is passed on to the virtual switch.

ESX Server ensures that frames are contained within the appropriate VLAN on a virtual switch. It does so in the following ways:

- VLAN data is carried outside the frame as it passes through the virtual switch. Filtering is a simple integer comparison. This is really just a special case of the general principle that the system should not trust user accessible data.
- Virtual switches have no dynamic trunking support.

Dynamic trunking and native VLAN are features in which an attacker may find vulnerabilities that could open isolation leaks. This is not to say that these features are inherently insecure, but even if they are implemented securely, their complexity may lead to misconfiguration and open an attack vector.

VLANs in VMware Infrastructure

VLANs provide for logical groupings of stations or switch ports, allowing communications as if all stations or ports were on the same physical LAN segment. Confining broadcast traffic to a subset of the switch ports or end users saves significant amounts of network bandwidth and processor time.

In order to support VLANs for VMware Infrastructure users, one of the elements on the virtual or physical network has to tag the Ethernet frames with 802.1Q tag, as shown in Figure 3. There are three different configuration modes to tag (and untag) the packets for virtual machine frames.

- Virtual switch tagging (VST mode) — This is the most common configuration. In this mode, you provision one port group on a virtual switch for each VLAN, then attach the virtual machine's virtual adapter to the port group instead of the virtual switch directly. The virtual switch port group tags all outbound frames and removes tags for all inbound frames. It also ensures that frames on one VLAN do not leak into a different VLAN.

Use of this mode requires that the physical switch provide a trunk.

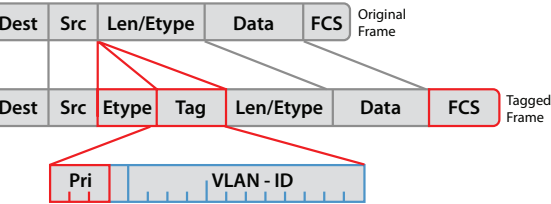


Figure 3 — Header of a packet tagged by an 802.1Q VLAN trunking driver.

- Virtual machine guest tagging (VGT mode) — You may install an 802.1Q VLAN trunking driver inside the virtual machine, and tags will be preserved between the virtual machine networking stack and external switch when frames are passed from or to virtual switches. The format for the header of a packet tagged in this way is shown in Figure 3.

Use of this mode requires that the physical switch provide a trunk.

- External switch tagging (EST mode) — You may use external switches for VLAN tagging. This is similar to a physical network, and VLAN configuration is normally transparent to each individual physical server.

There is no need to provide a trunk in these environments.

For details on using VLANs with VMware Infrastructure, see the white paper titled VMware ESX Server 3 802.1Q VLAN Solutions, available from the VMTN Web site (<http://www.vmware.com/vmtn/>).

NIC Teaming

You can connect a single virtual switch to multiple physical Ethernet adapters using the VMware Infrastructure feature called NIC teaming. A team can share the load of traffic between physical and virtual networks among some or all of its members and provide passive failover in the event of a hardware failure or a network outage. You can set NIC teaming policies at the port group level.

Note: All physical switch ports in the same team must be in the same Layer 2 broadcast domain.

Load Balancing

Load balancing allows you to spread network traffic from virtual machines on a virtual switch across two or more physical Ethernet adapters, giving higher throughput than a single physical adapter could provide. When you set NIC teaming policies, you have the following options for load balancing:

- **Route based on the originating virtual switch port ID** — Choose an uplink based on the virtual port where the traffic entered the virtual switch. This is the default configuration and the one most commonly deployed.

When you use this setting, traffic from a given virtual Ethernet adapter is consistently sent to the same physical adapter unless there is a failover to another adapter in the NIC team.

Replies are received on the same physical adapter as the physical switch learns the port association.

This setting provides an even distribution of traffic if the number of virtual Ethernet adapters is greater than the number of physical adapters.

A given virtual machine cannot use more than one physical Ethernet adapter at any given time unless it has multiple virtual adapters.

This setting places slightly less load on the ESX Server host than the MAC hash setting.

Note: If you select either srcPortID or srcMAC hash, you should not configure the physical switch ports as any type of team or bonded group.

- **Route based on source MAC hash** — Choose an uplink based on a hash of the source Ethernet MAC address.

When you use this setting, traffic from a given virtual Ethernet adapter is consistently sent to the same physical adapter unless there is a failover to another adapter in the NIC team.

Replies are received on the same physical adapter as the physical switch learns the port association.

This setting provides an even distribution of traffic if the number of virtual Ethernet adapters is greater than the number of physical adapters.

A given virtual machine cannot use more than one physical Ethernet adapter at any given time unless it uses multiple source MAC addresses for traffic it sends.

- **Route based on IP hash** — Choose an uplink based on a hash of the source and destination IP addresses of each packet. (For non-IP packets, whatever is at those offsets is used to compute the hash.)

Evenness of traffic distribution depends on the number of TCP/IP sessions to unique destinations. There is no benefit for bulk transfer between a single pair of hosts.

You can use link aggregation — grouping multiple physical adapters to create a fast network pipe for a single virtual adapter in a virtual machine.

When you configure the system to use link aggregation, packet reflections are prevented because aggregated ports do not retransmit broadcast or multicast traffic.

The physical switch sees the client MAC address on multiple ports. There is no way to predict which physical Ethernet adapter will receive inbound traffic.

All adapters in the NIC team must be attached to the same physical switch or an appropriate set of stacked physical switches. (Contact your switch vendor to find out whether 802.3ad teaming is supported across multiple stacked chassis.) That switch or set of stacked switches must be 802.3ad-compliant and configured to use that link-aggregation standard in static mode (that is, with no LACP). All adapters must be active. You should make the setting on the virtual switch and ensure that it is inherited by all port groups within that virtual switch.

Failover Configurations

When you configure network failover detection you specify which of the following methods to use for failover detection:

- **Link Status only** — Relies solely on the link status provided by the network adapter. This detects failures, such as cable pulls and physical switch power failures, but it cannot detect configuration errors, such as a physical switch port being blocked by spanning tree or misconfigured to the wrong VLAN or cable pulls on the other side of a physical switch.
- **Beacon Probing** — Sends out and listens for beacon probes — Ethernet broadcast frames sent by physical adapters to detect upstream network connection failures — on all physical Ethernet adapters in the team, as shown in Figure 4. It uses this information, in addition to link status, to determine link failure. This detects many of the failures mentioned above that are not detected by link status alone, however beacon probing should not be used as a substitute for a robust redundant Layer 2 network design. Beacon probing is most useful to detect failures in the closest switch to the ESX Server hosts, where the failure does not cause a link-down event for the host.

By default, NIC teaming applies a fail-back policy. That is, if a physical Ethernet adapter that had failed comes back online, the adapter is returned to active duty immediately, displacing the standby adapter that took over its slot. This policy is in effect when the Rolling Failover setting is No.

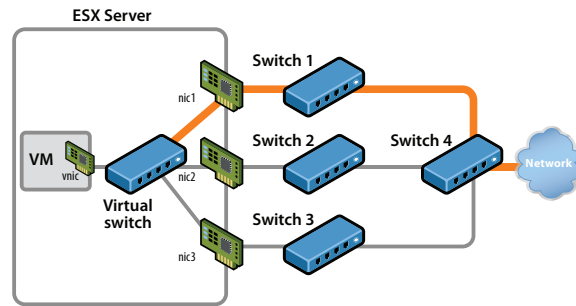
If the primary physical adapter is experiencing intermittent failures, this setting can lead to frequent changes in the adapter in use. The physical switch thus sees frequent changes in MAC address, and the physical switch port may not accept traffic immediately when a particular adapter comes online. To minimize delays, disable the following on the physical switch:

- Spanning tree protocol (STP) — disable STP on physical network interfaces connected to the ESX Server host. For Cisco-based networks, enable port fast mode for access interfaces or portfast trunk mode for trunk interfaces (saves about 30 seconds during initialization of the physical switch port).
- Etherchannel negotiation, such as PAGP or LACP — must be disabled because they are not supported.
- Trunking negotiation (saves about four seconds).

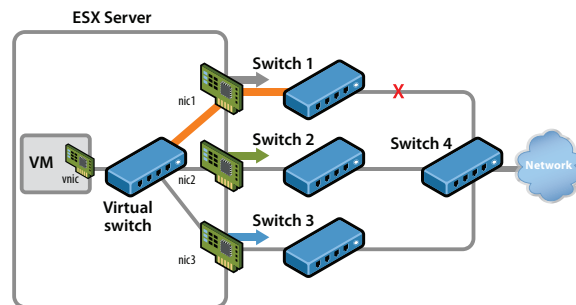
An alternative approach is to set Rolling Failover to Yes. With this setting, a failed adapter is left inactive even after recovery until another currently active adapter fails, requiring its replacement.

Using the Failover Order policy setting, you specify how to distribute the work load for the physical Ethernet adapters on the host. You can place some adapters in active use; designate a

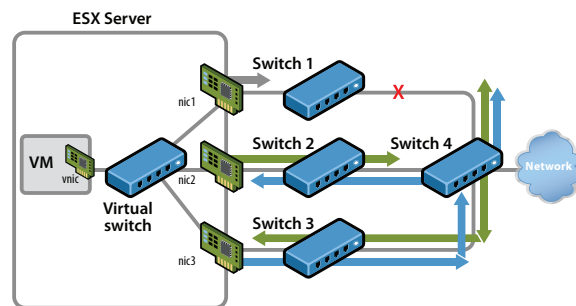
second group as standby adapters for use in failover situations; and designate other adapters as unused, excluding them from NIC teaming.



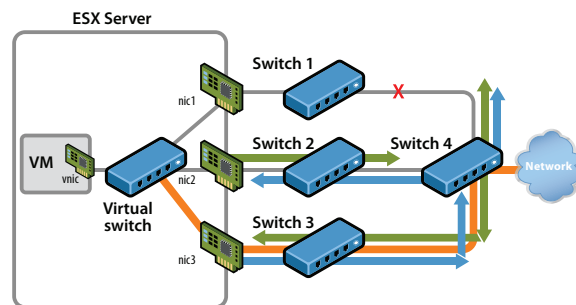
Virtual machine network traffic handled by nic1



Connection between switch 1 and switch 4 fails
Each Ethernet adapter sends a beacon packet



Beacons returned on nic2 and nic3, not returned on nic1



Data to network rerouted through nic3

Figure 4 — Using beacons to detect upstream network connection failures.

Using the Notify Switches policy setting, you determine how ESX Server communicates with the physical switch in the event of a failover. If you select Yes, whenever a virtual Ethernet adapter is connected to the virtual switch or whenever that virtual Ethernet adapter's traffic would be routed over a different physical Ethernet adapter in the team due to a failover event, a notification is sent out over the network to update the lookup tables on physical switches. In almost all cases, this is desirable for the lowest latency when a failover occurs.

Note: If you are using Microsoft Network Load Balancing in unicast mode, do not set Notify Switches to Yes. No such issue exists with NLB running in multicast mode, the mode VMware strongly recommends.

Layer 2 Security Features

The virtual switch has the ability to enforce security policies to prevent virtual machines from impersonating other nodes on the network. There are three components to this feature.

- Promiscuous mode is disabled by default for all virtual machines. This prevents them from seeing unicast traffic to other nodes on the network.

- MAC address change lockdown prevents virtual machines from changing their own unicast addresses. This also prevents them from seeing unicast traffic to other nodes on the network, blocking a potential security vulnerability that is similar to but narrower than promiscuous mode.
- Forged transmit blocking, when you enable it, prevents virtual machines from sending traffic that appears to come from nodes on the network other than themselves

Managing the Virtual Network

VMware VirtualCenter provides tools for building and maintaining your virtual network infrastructure, as shown in Figure 5.

You can use VirtualCenter to add, delete, and modify virtual switches and to configure port groups with VLANs and teaming.

You can use the VirtualCenter roles feature to assign the permissions a network administrator needs to manage the virtual network. For a more detailed discussion, see the paper “Managing VMware VirtualCenter Roles and Permissions,” available at <http://www.vmware.com/vmtn/resources/826>.

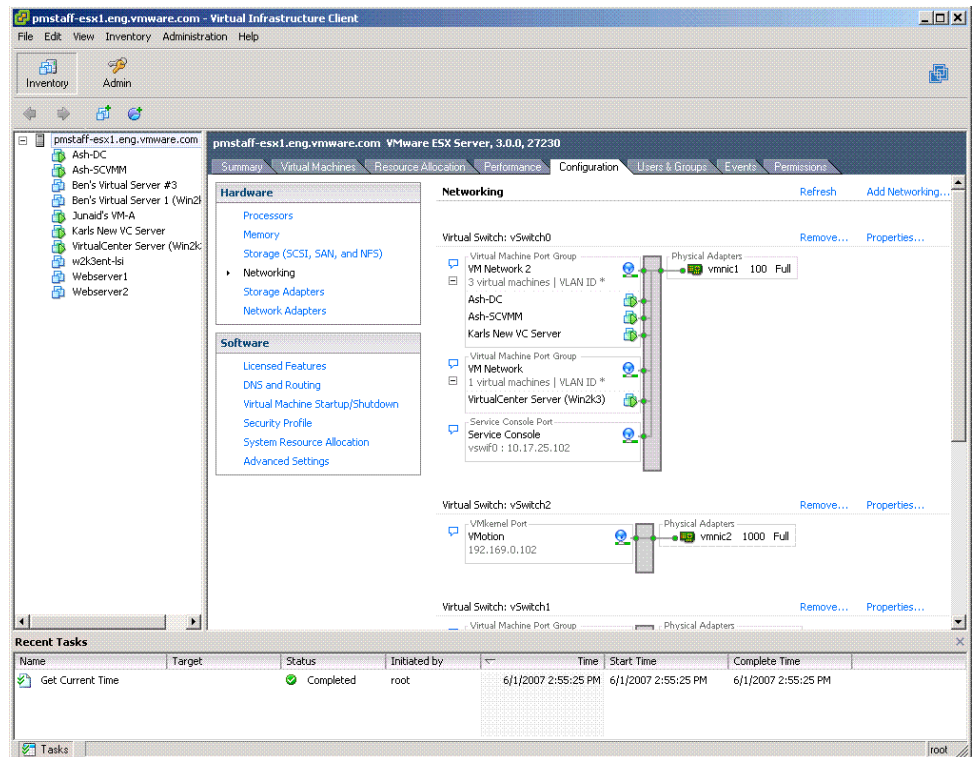


Figure 5 — Using VMware VirtualCenter to manage a virtual network.

Appendix: Virtual Device Maximums

The following table summarizes the maximum numbers of various virtual and physical networking devices that can be configured using VMware Infrastructure 3 (accurate for ESX Server 3.0 and ESX Server 3.0.1).

Device	Maximum Number
Virtual Ethernet adapters per virtual machine	4
Virtual switch ports per host	4096
Virtual switch ports per switch	1016
Virtual switches per host	248
Uplinks per virtual switch	32
Uplinks per host	32
Virtual switch port groups per host	512
Physical e1000 Ethernet adapters per host	32 (maximum tested)
Physical Broadcom Ethernet adapters per host	20 (maximum tested)
Physical e100 Ethernet adapters per host	26 (maximum tested)

Revision: 20070718 Item: IN-018-INF-01-01



VMware, Inc. 3401 Hillview Ave. Palo Alto CA 94304 USA Tel 650-475-5000 Fax 650-475-5001 www.vmware.com

© 2007 VMware, Inc. All rights reserved. Protected by one or more of U.S. Patent Nos. 6,397,242, 6,496,847, 6,704,925, 6,711,672, 6,725,289, 6,735,601, 6,785,886, 6,789,156, 6,795,966, 6,880,022, 6,961,941, 6,961,806, 6,944,699, 7,069,413; 7,082,598, 7,089,377, 7,111,086, 7,111,145, 7,117,481, 7,149,843, 7,155,558, and 7,222,221; patents pending.

VMware, the VMware "boxes" logo and design, Virtual SMP and VMotion are registered trademarks or trademarks of VMware, Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.

