VMware® Infrastructure 3

Advanced Technical Design Guide

~and~

Advanced Operations Guide

Two books in one!

Ron Oglesby
Scott Herold
Mike Laverick

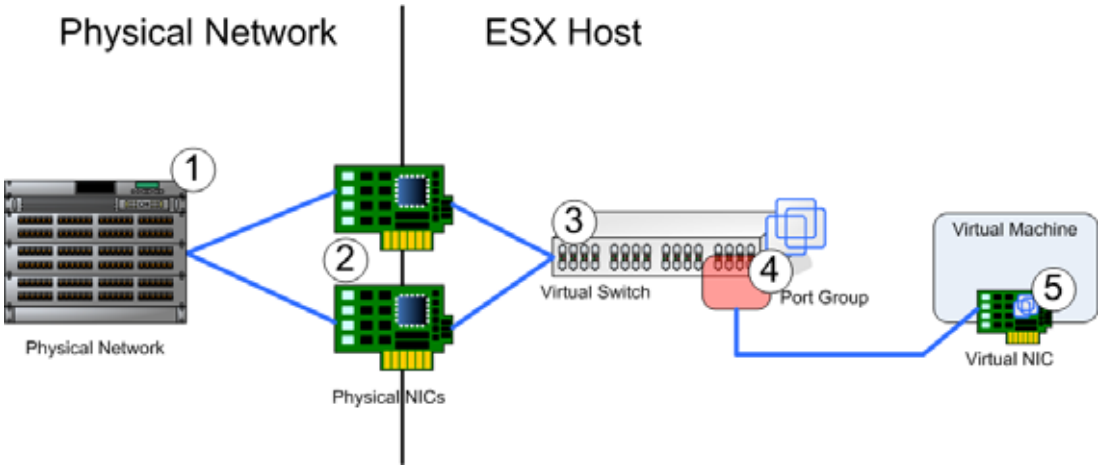
Chapter 6 - Networking Concepts and Strategies

We discussed some very high level basics of networking in an ESX infrastructure back in Chapter 2. This chapter is going to highlight the key concepts and strategies surrounding the networking capabilities of the virtual infrastructure. Here we are going to discuss the advanced networking aspects of the physical hardware, virtual machines, and network based storage configurations. The end of this chapter will focus on several usage scenarios that are common in many enterprise virtual infrastructures.

Key Concepts of ESX Networking

Here we want to start by referring back to the image from Chapter 2 and discuss in more detail on how each of the various network components of a virtual infrastructure are used.

Figure 6- 1: Network Components



Physical Connectivity

Physical network connectivity is not required to actually run and leverage virtual machines. Having physical network connectivity will make it possible to re-

motely manage the infrastructure and will allow external systems to interact virtual machines running within the environment. We suspect that there is no reason for anyone reading this book to not plug an ESX host into the network and will focus on what it takes to get connected.

Physical Network

While physical network components such as switches and routers are not a direct component of ESX, we feel it is important to discuss some of the design options surrounding the configuration and use of a redundant network switch infrastructure. VMware provides several mechanisms of load balancing for configuring network components, each of which inherently provide redundancy when connected to multiple physical switches. Further details surrounding these mechanisms will appear a little later in this chapter. Knowing this information it is important to review the physical network layout to determine if it is possible to connect your physical NICs to multiple physical switches.

If connecting to multiple physical switches is not possible it is recommended that multiple blades of a switch chassis be used. This will at lease provide protection if a particular switch blade fails. Failing that, there is nothing wrong with using a single physical switch for all network connectivity for your virtual infrastructure, you just need to be aware of the implications of introducing a single point of failure into the environment.

When configuring the switch ports that you will connect your ESX hosts to you must be aware that the Spanning-Tree protocol has been known to cause delays in re-establishing network connectivity after a failed link returns to service. It is not uncommon to see 15-20 ICMP requests fail when a physical NIC within a virtual switch re-establishes its connection after a failure. To minimize the impact of the Spanning-Tree protocol, there are two things that can potentially be configured at the physical switch. The first option is disabling the Spanning-Tree protocol for the entire physical switch. There is an extremely good chance that this is not possible in most environments. The alternate, and most likely solution to resolve some of these issues would be to enable Portfast Mode for the ports that the physical NICs are connected to.

Every hardware vendor is unique and has different ways to configure various settings. We recommend working with the vendors closely if there are questions in regards to a specific configuration that is required for VMware compatibility.

Physical NICs

It should go without saying that physical network adapters are required in order to enable remote communication to not only the ESX hosts, but also the virtual machines they contain. Pretty much any modern network card that is either embedded on a server, or that can be added as a PCI card, is compatible with ESX Server. We are finally in a time where 100Mb networking is coming to an end and most organizations have implemented a gigabit network infrastructure. When you start taking a look at the nature of ESX and its intended implementation, 100Mb just doesn't cut it anymore…for any physical connection on the ESX host. Before attempting to setup your ESX host you should verify with either VMware or you hardware vendor to ensure the chosen adapters are fully compatible for ESX 3.

The number of physical network adapters installed in a system is dependent on several factors such as the level of redundancy required or the desired VLAN configuration of your solution. A typical configuration will optimally contain anywhere from 4 to 5 physical network adapters for each ESX host in the solution. It is possible to have an acceptable solution with 3 physical connections, but having only 2 is definitely not recommended. More than 5 may be required if the politics of the office get in the way of allowing the ESX server admins to take ownership of VLAN management and configuring VLAN tagging is not possible on the ESX hosts. We will discuss the use cases for various NIC configurations at the end of this chapter.

VMware has put limitations on the number of physical NICs that may be installed on a single physical host running ESX 3. While these are hard maximums, anyone who comes remotely close to half of these limits are encouraged to call a VMware service provider to show you how to actually build a virtual infrastructure. The reason behind variations between NICs typically has to do with the amount of memory required by the necessary drivers of each individual device.

Device	Count
Intel e100	26
Intel e1000	32
Broadcom	20

Virtual Connectivity

Virtual connectivity of your virtual machines, more often than not, is quite a bit more complicated than setting up the physical networking surrounding the infrastructure. It is actually possible to construct a fully functional enterprise network infrastructure using the logical components provided by VMware ESX. The three key concepts that we will focus on for this section are the use and configuration of Virtual NICs, Virtual Switches, and Port Groups.

Virtual NICs

A Virtual NIC (vNIC) is a network adapter that is configured within ESX for use by a virtual machine. Each virtual machine may have up to 4 virtual NICs configured, providing significant amounts of flexibility when connecting a virtual server to multiple subnets. Each vNIC that is assigned to a virtual machine receives its own unique MAC address, just as a physical adapter in a physical server would. In fact, if you were to look at the configuration of the network ports on your physical switch you would see that these MAC addresses are being published all the way into the physical infrastructure.

The way that these virtual NICs are presented to your virtual machine is actually done one of several different ways. By default, any newly constructed virtual machine that is configured with a virtual NIC is configured with the "Flexible" VMware adapter. The theory behind this adapter differs slightly, and improves on a design flaw from previous versions of ESX Server.

The Flexible adapter has the capabilities of both the legacy VLANCE and VMXNET drivers from previous versions of ESX rolled up into a single device.

It has the basic compatibility that is required by older operating systems, and with the proper VMware drivers installed, has the compatibility and advanced feature set that the VMXNET device used to provide.

The addressed design flaw with this solution was the fact that in legacy operating systems if you configured the system with the VLANCE driver and decided to change this to VMXNET at a later time, it was a device modification that was reported to the operating system. IP Address settings and device naming would not carry over during this change. By combining the functionality of both devices into this Flexible adapter the only modification required is a driver installation (typically with the VMware tools) in the guest operating system and the full feature set is instantly enabled without having a new device added to the system.

When working with 64-Bit operating systems VMware will instruct the guest to an Intel e1000 device driver. While it is possible to manually run 32bit operating systems with the e1000 device by modifying the VMX file of the virtual machine, it may not be officially supported by VMware, and full functionality may be questionable. If you will be running 64bit operating systems you will have no choice but to use this driver.

If using either the Flexible driver in VLANCE mode or the e1000 driver, ESX will be emulating their respective adapters. This will incur some virtualization overhead in the fact that every network call must be translated by the hypervisor to properly communicate. The Flexible driver, while running in VMXNET mode is a paravirtualized device, which means it is tightly integrated with the VMkernel's network stack and does not incur as much overhead as a strictly emulated device.

We mentioned that each virtual NIC that is configured for use with a virtual machine receives a unique MAC address. Fortunately VMware has gone through the process of registering their OUIs with the IEEE. In theory, you should never see a VMware MAC address outside one of the following ranges:

00:05:69
00:0c:29
00:1c:14
00:50:56

Virtual Switches

A virtual switch is a logical component of ESX that acts identically to a physical layer 2 switch and is used to map a Virtual NIC back to individual or groups of Physical NICs for network connectivity. In addition, virtual switches can be configured in a "private" mode that has no connectivity to the physical network, but still allows virtual machines to communicate internally to the ESX host.

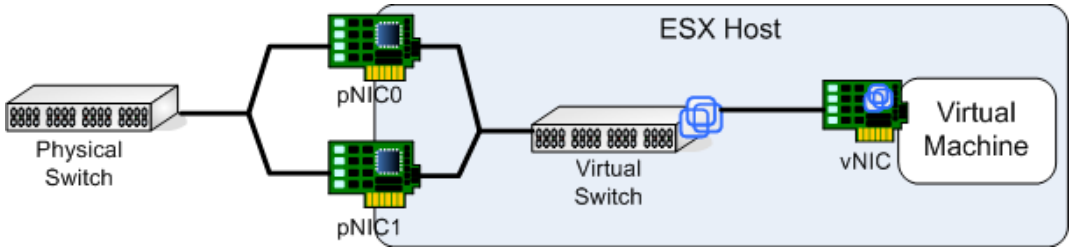
Whether you are configuring your virtual switch in a public or private mode, there are several functions that are identical. Each virtual switch can be configured to support up to 1016 vNICs. This is a FAR cry better than the 32 vNICs supported in previous versions of ESX. Traffic that flows between any two vNICs on the same virtual switch will actually transfer over the system bus and will not traverse the network (assuming VLAN tagging is not being utilized, but more on that in a moment). When laying out your network design, this should be considered, as it is often advantageous to have network intensive applications talking directly to each other. When information is transferred in this manner, no network resources are impacted. All processing in this scenario is handled by processor 0 of the ESX host system. As neat as this sounds, there is a down side in the fact that if your traffic never hits the wire, typical networking monitoring and packet sniffing technologies will not be able to see the communication between VMs.

One key fact to remember is that there is a hard limit of 127 virtual switches allowed on an ESX host. This, like most of VMware's hard limits should not come into play in most normal situations. The only chance of seeing any configuration ever coming close to this limit is if an ESX host is being used alongside a VMware Lab Manager implementation.

Public Virtual Switches

Public virtual switches are easily the most utilized virtual switches in an ESX environment. In a public virtual switch, anywhere from 1 to 32 (Depending on physical hardware) Physical NICs may be bound, providing connectivity to the physical network for virtual machines. When utilizing a single Physical NIC, it is important to remember that there will be no redundancy in the network design for your virtual machines. It is also important to note that a Physical NIC can only belong to one virtual switch at a time.

Figure 6- 2: Public Virtual Switch

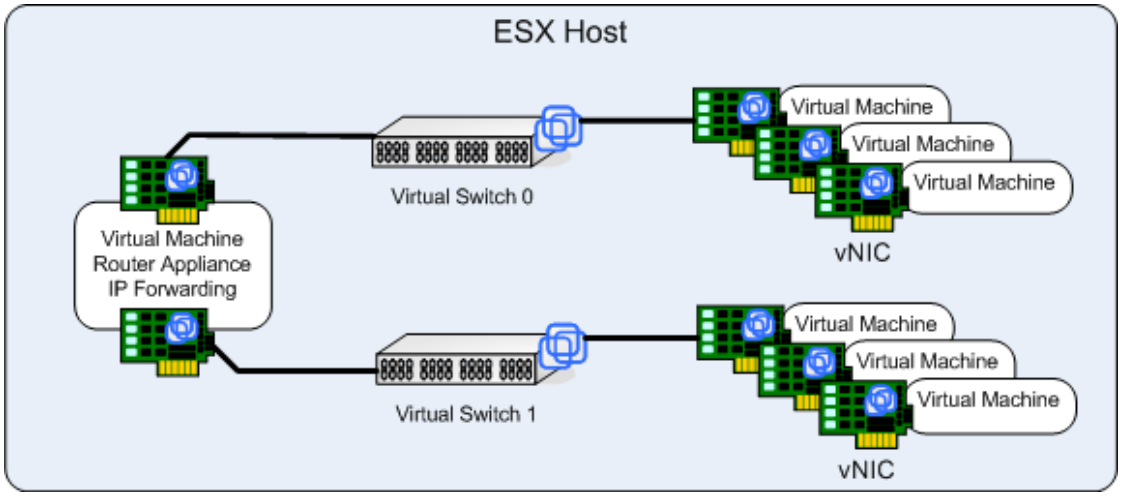


In a default configuration, all Physical NICs are configured in a redundant bond that load balances virtual machines across all adapters in that bond. We will discuss the various load balancing mechanisms later in this chapter. In addition to configuring load balancing methods, each virtual switch may be configured to handle VLAN configurations in different ways. By default, all virtual machines become extensions of the VLAN that the physical switch port is configured for. ESX provides mechanisms to handle VLANs in different manners. Alternative VLAN configurations will also be reviewed at a later time in this chapter.

Private Virtual Switches

The main difference between public virtual switches and private virtual switches lies in the fact that private virtual switches do not have a connection to the physical network. In a private virtual switch, all traffic generated between virtual machines is handled by the CPU and transferred over the system bus, as there is no other way for this traffic to travel between virtual machines. Private virtual switches provide isolation and can serve several purposes such as testing AD schema changes with a namespace that is identical to production, or building a DMZ environment in a box.

Figure 6- 3: Private Virtual Switch



Virtual machines that are connected to a private virtual switch must be configured for the same subnet in order to communicate with each other. Multiple private virtual switches can be created within an ESX host to provide a multiple subnet environment. ESX does not provide routing capabilities internally across virtual switches. The only way that virtual machines that are on different virtual switches can communicate is by configuring a guest operating system with IP forwarding enabled. Guest operating systems must then have either static routes defined or default gateways configured that point to this "gateway" server to communicate across subnets. There are also several virtual machine appliances available for download that provide a lightweight Linux kernel with routing capabilities built in.

Port Configuration

The default configuration of a virtual switch is to enable 56 ports. Each Virtual NIC that is assigned to a virtual machine takes up one of these ports…not unlike a physical infrastructure. VMware has extended the capabilities of ESX 3 to allow significantly more ports than previous versions. Within the Virtual Infrastructure Client it is possible to configure a virtual switch to contain 8, 24, 56, 120, 248, 504, or 1016 virtual switch ports. Modifying this value does require a reboot of the ESX host to take effect, so plan accordingly. Also, if you do plan on using more than the default 56 allotted ports, consider increasing the service console memory from its default value.

Virtual Switch Configuration Options

Virtual switches, while not as intelligent as a typical physical switch, do have some configurable intelligence built in. The configuration of the virtual switch provides flexibility to the end user to provide some advanced functionality in security, bandwidth throttling, and redundancy settings.

Security

Virtual switches may be configured with several Layer 2 security policies. These configurations provide useful measures to enabled (or disable) typical configurations that can go above and beyond what can normally be done in a strictly physical environment.

Promiscuous Mode – Enabling promiscuous mode enables a virtual switch to receive all frames, not just those targeted for the virtual NICs configured within that switch. This is disabled by default

MAC Address Changes – Disabling this prevents guest virtual machines from changing the MAC address to something other than what is specified in its VMX file. This is enabled by default.

Forged Transmits – Disabling this prevents guest virtual machines from modifying the source MAC address of a frame to something other than what is currently registered on the virtual switch for that guest. This prevents IP Spoofing mechanisms. The default configuration is to allow Forged Transmits.

Traffic Shaping

Within ESX 3, it is also possible to limit the OUTGOING bandwidth through a virtual switch. It is very important to note that there is no way to limit the amount of bandwidth coming into a virtual switch. There are several basic configurations in regards to setting up bandwidth throttling.

Average Bandwidth - This setting is the sustained throughput that you would like the virtual switch to maintain for its guests.

Peak Bandwidth - This is the maximum amount of throughput allowed through a virtual switch. A virtual switch may hit its peak bandwidth to assist in properly processing all data that needs to be sent. Peak bandwidth is often configured to double the value of average bandwidth. This allows the virtual machines to properly send all packets when the system comes under load.

Burst Size - The amount of data that may be sent through a virtual switch while hitting its peak bandwidth. If the burst amount is hit, the virtual switch will drop below the peak bandwidth until the next burst cycle is available. The burst size should be configured as a byte value of 15% of the average bandwidth. This should be more than enough throughput to properly fill the peak bandwidth rate.

If the application or guest operating system being filtered starts to display errors such as dropped connections or failed ICMP requests, you should consider increasing these values accordingly.

NIC Teaming

The NIC Teaming functionality that ESX 3 provides is one of the most powerful features of the virtual switch architecture. The scenarios that may be configured through the use of the various teaming functionality provides unparalleled levels of high availability and load balancing for your virtual infrastructure.

Load Balancing

VMware ESX provides three methods for load balancing the traffic generated by virtual machines. Each method, in addition to providing load balancing for virtual switch traffic, assists in providing a redundant network design. There is also a fourth method of redundancy in which a Physical NIC within a virtual switch may be configured in a standby mode, but this does not provide any load balancing.

Virtual Port Based – When a virtual NIC is assigned to a virtual switch, it is also assigned a virtual port ID that is remembered by the virtual switch. Using Virtual Port Based load balancing, which is the default setting of VI3, leverages a round robin mechanism to assign virtual port IDs down an active physical

adapter of the virtual switch. Once this virtual port is assigned to a physical NIC it will not change when the system is rebooted or goes offline. New virtual port IDs that are created are assigned to the next available physical NIC, even if it is only for temporary use.

Over time, it is possible for virtual switches to have an uneven number of virtual machines down each path, but it is significantly less of an issue than using the MAC Based mechanism that we will discuss next. Since virtual port IDs are simply assigned to the next active physical NIC, there is almost no processing or logic behind the assignment, which has the smallest impact to ESX host for managing failover. The VMkernel is responsible for managing which physical NIC in a virtual switch a virtual port, and subsequently, a virtual MAC address of a virtual NIC, will communicate with. The VMkernel will only announce the MAC address of the virtual machine down the active path to prevent issues where duplicate MAC addresses are broadcast to every physical switch port. This also allows the VMkernel to manage failover if a link failure is detected. The VMkernel will send a new ARP request down the new active path to reestablish communication to the virtual machine. In fact, with Virtual Port Based load balancing, physical NICs bound to a virtual switch may span across multiple physical switches for maximum fault tolerance.

MAC Based - MAC Address Load Balancing was the default load balancing mode for previous versions of ESX Server. Like Virtual Port Based balancing, this method does not require that any additional switch configuration be made to the physical switches that ESX is connected to, making it an ideal candidate in terms of network infrastructure compatibility. In this load balancing mode, the VMkernel has full control over which physical NIC in a virtual switch publishes the virtual machine's MAC address to the physical switch infrastructure through the use of a calculated hash based on the actual MAC address. By only allowing one physical NIC to announce a MAC address for a virtual machine, there are no duplicate MAC issues on the physical switches that prevent the virtual machine from properly communicating. Like Virtual Port Based load balancing, MAC Based configurations may span across multiple physical switches. MAC Based load balancing is introduced in VI3 for legacy and compatibility purposes. It is recommended that you use Virtual Port Based load balancing for all but the most intensive workloads.

While Virtual Port and MAC Address Load Balancing may be the easiest of the three methods to set up, they are not the most efficient at load balancing traffic.

The VMkernel uses an internal algorithm to determine which physical NIC in a virtual switch a specific virtual MAC address gets announced through. It is not possible to manually configure which virtual machines communicate down specific paths of a virtual switch. What this means is that the VMkernel simply tells a specific physical NIC to handle the traffic for a virtual NIC without regard to the amount of traffic being generated. We have seen instances on ESX where one physical NIC is generating significantly more traffic than any other within a virtual switch. When we do ESX designs in the field we do not utilize MAC Address Load Balancing to actually load balance traffic; it is used as a redundancy method and we use it when we know we have the appropriate network capacity within a virtual switch to handle an N+1 configuration. If we lose one physical NIC in a virtual switch, we should still have the appropriate capacity to provide the virtual machines with the throughput they need for operation.

IP Based - The third method that ESX is capable of providing for load balancing is based on destination IP address. Since outgoing virtual machine traffic is balanced based on the destination IP address of the packet, this method provides a much more balanced configuration than Virtual Port or MAC Address based balancing. Like the previous methods, if a link failure is detected by the VMkernel, there will be no impact to the connectivity of the virtual machines. The downside of utilizing this method of load balancing is that it requires additional configuration of the physical network equipment.

Because of the way the outgoing traffic traverses the network in an IP Address load balancing configuration, the MAC addresses of the virtual NICs will be seen by multiple switch ports. In order to get around this "issue", either EtherChannel (assuming Cisco switches are utilized) or 802.3ad (LACP - Link Aggregation Control Protocol) must be configured on the physical switches. Without this configuration, the duplicate MAC address will cause significant switching issues. This option is only required for the most intense of network workloads, which is not very often. It is surprisingly difficult to truly max out the throughput of a properly configured ESX host based on our configuration recommendations at the end of this chapter. If you don't need to introduce this much complexity into the networking environment of your virtual infrastructure, don't use this option.

Use Explicit Failover Order – As we mentioned, there was a fourth "load balancing" option that can be configured for a virtual switch. This option is simply, don't load balance. Instead this instructs the virtual switch to simply use

the highest priority active NIC for the switch and simply "failover" down the list of alternate active NICs.

Network Failover Detection

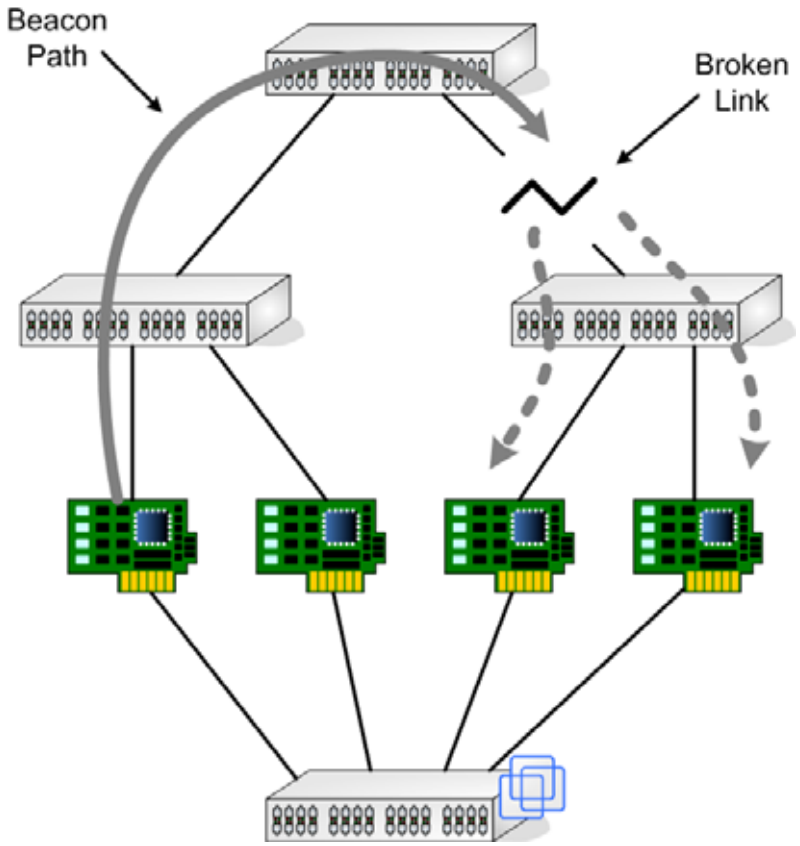
VMware ESX 3 provides two different methods for determining whether a link to a physical NIC has failed. The first method is simply if the physical NIC detects that the link between it and its uplink physical switch has failed. This works for most scenarios that ESX may be configured in.

There is an alternate method that may be used that addresses the situation of "What if my uplink switch is still active and online, but it cannot communicate with ITS uplink switch. As far as ESX is concerned, the link is active and the virtual switch stays online. The problem comes in when a node that isn't attached to that same physical switch requires access to one of the virtual machines.

Beacon monitoring provides protection in cases such as these. If you have properly configured your network, you will be using multiple uplink switches per virtual switch to eliminate any single points of failure. In the event a single port or cable fails in the core switch, the second physical switch can still manage the load of the virtual machines. The only way to detect the failure between the uplink switches and their core switch is by enabling beacon monitoring.

The idea here is that ESX Server will send beacon packets from one physical NIC to all other physical NICs in the same virtual switch. The beacon packets will then traverse the network through the switch infrastructure to the other physical NICs on the virtual switch. If there is no response from beacon packets, this means there is a network disruption somewhere upstream.

Figure 6- 4: Beacon Monitoring



Using beaconing is only effective when the Physical NICs of a virtual switch are connected to different uplink switches. If the Physical NICs are only connected to a single uplink switch, then beaconing does you no good since you are adding beaconing overhead to traverse a path between switch ports on the same physical device. Using beaconing on a single physical switch will tell you nothing more than the fact that the device or its ports are still active, which ESX server will handle by default, without beaconing.

Unless you absolutely need to have ESX monitor your upstream network we would veer away from it. Beacon monitoring sometimes detects network failures that are not really happening.

Port Groups

The final component of the virtual network infrastructure is also one of the most important. Port groups are logical groups of virtual switch ports that

share common settings such as Security, Traffic Shaping, Load Balancing, and VLAN configurations. Up to 512 port groups may be configured on any given ESX host at a time. There are three different types of port groups that may be configured on an ESX host

Service Console

Service Console port groups are used to configure access to the ESX service console for management of the physical host. Service Console port groups may not be used for any other purpose than providing access as a "vswif" device in the Linux management console. Each vswif device that is formed by creating a Service Console port group can be configured with a unique IP Address.

VMkernel

VMkernel port groups are used to configure either VMotion or network based storage access for NFS or iSCSI volumes. VMkernel port groups cannot be used to run virtual machines or allow access to the service console. A VMkernel port group that is to be used for VMotion must be explicitly configured to allow VMotion access. An IP Address must be properly configured at the VMkernel level to provide access to either VMotion functionality or to enable communication with network based storage devices.

Virtual Machine

Virtual Machine port groups are those that allow network connectivity for your virtual machines, whether through a public or private virtual switch. Every virtual NIC that is created within a virtual machine must be assigned to a virtual machine port group. Each of these virtual machine port groups, like all other port groups, contains unique settings. In the case of virtual machine port groups, different VLAN tagging configurations, as you will find out very shortly, may be assigned to various port groups. Each virtual NIC may be assigned to only one port group at a time, but a single virtual machine may have up to 4 different virtual NICs, each in a unique port group configuration. These virtual machine port groups are leveraged to define unique identities to the various network configurations for your virtual machine environment.

Virtual machine port groups are not configured with an IP Address. Instead, they either become an extension of the VLAN of the physical switch port they are connected to, or can be configured as trunk ports

Naming Standards

Port Groups are the single networking component that allows user specified names. Careful planning should go into the creation of a proper naming standard for port groups. Quickly determine the difference between "Port Group 1", "Port Group 2", and "Port Group 3" when building a virtual machine could be a very tedious process.

Implementations of ESX Server in a small or medium sized business could be as simple of a process as naming your virtual switches "Production Network", "Development Network", and "Quarantine". It is very easy, with a very quick glance, to determine the difference between these port groups with a proper naming convention.

Enterprise organizations are a difference story. Often times there are hundreds or even thousands of "Production Networks", so such a simplistic naming standard will probably not work. In these cases it often comes down to integrating naming conventions into already existing network IT policies. If applications serve as network boundaries, it may be as simple as calling your port group "Application1 Prod", and "Application1 Dev". In environments where the network is simply grown as new systems are implemented, simply naming your port groups after VLAN configurations may be sufficient. Simple names such as "VLAN 102" and VLAN 786" often suffice.

When creating and naming port groups use some common sense and integrate these naming conventions with your everyday business practices to make something that is easily identifiable to anyone responsible for building virtual machines.

Port Group Configurations

It is possible to configure port groups with the same Virtual Switch configuration options. By default, a port group will automatically assume the configuration of the virtual switch it is created on. This may be overwritten so

each port group contains a unique configuration, although they belong ot the same virtual switch.

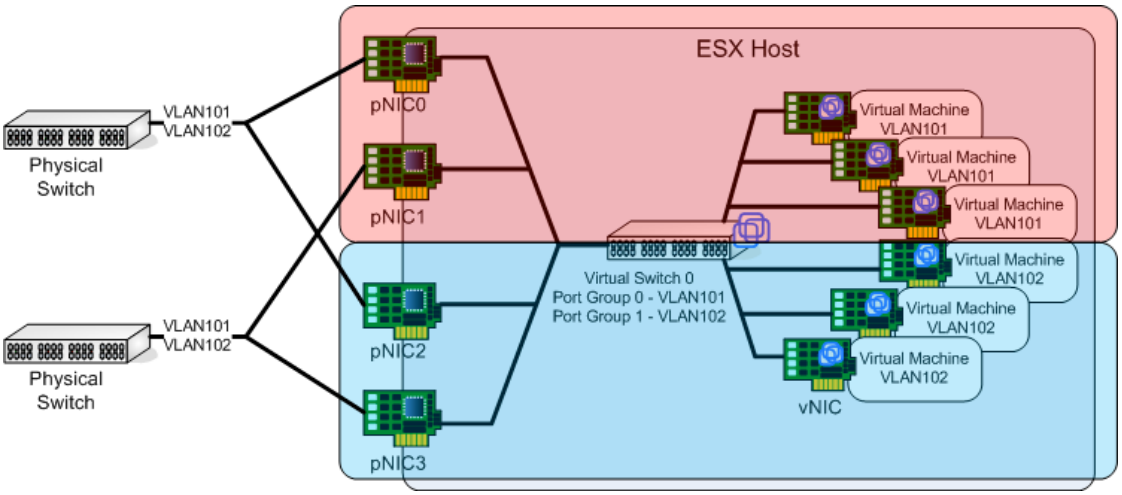
As an example, let's take a look at the following scenario represented by figure 6-5.

A virtual switch is created on an ESX host and is made up of 4 physical network adapters split across two physical switches. The virtual switch has no advanced configuration in place outside the default settings.

Two port groups are created, one for VLAN 101 and one for VLAN 102.

Each port group has two specific Physical NICs configured as Active NICs for the port group and two specific Physical NICs configured for failover purposes. Each physical NIC assigned as Active for the port group goes to a different physical switch.

Figure 6- 5: Port Groups



This solution is extremely redundant for several reasons. First, the virtual switch is made up of four physical NICs. Even under medium to heavy workloads the ESX host could lose up to two NICs and still have enough capacity to continue to run the virtual workload. Second, the port groups on the virtual switch are configured so that up to two failures at a minimum are required to bring the system down entirely. The port group has two physical NICs it can

use for balancing traffic, each connected to a different physical switch. Even if the two primary physical NICs go down the port group will share the failover NICs with the second port group. The largest, and most unlikely event that can bring the virtual machines down is if both physical switches fail at the same time. While this is not impossible, it is quite unlikely to happen except in a major disaster or business crippling event.

One thing it is very important to note is that when you are configuring your virtual switches and port groups with the intent of using VMotion, the configurations must match across the board on all ESX hosts in the cluster. This includes the proper naming of the port groups themselves. VirtualCenter will happily let you know that there are invalid settings across these required common components if a VMotion is attempted when improperly configured.

The single configuration option that is available to port group configurations vs. virtual switch configurations is in the fact that you can assign a port group to only listen to packets tagged with a particular VLAN.

VLANs

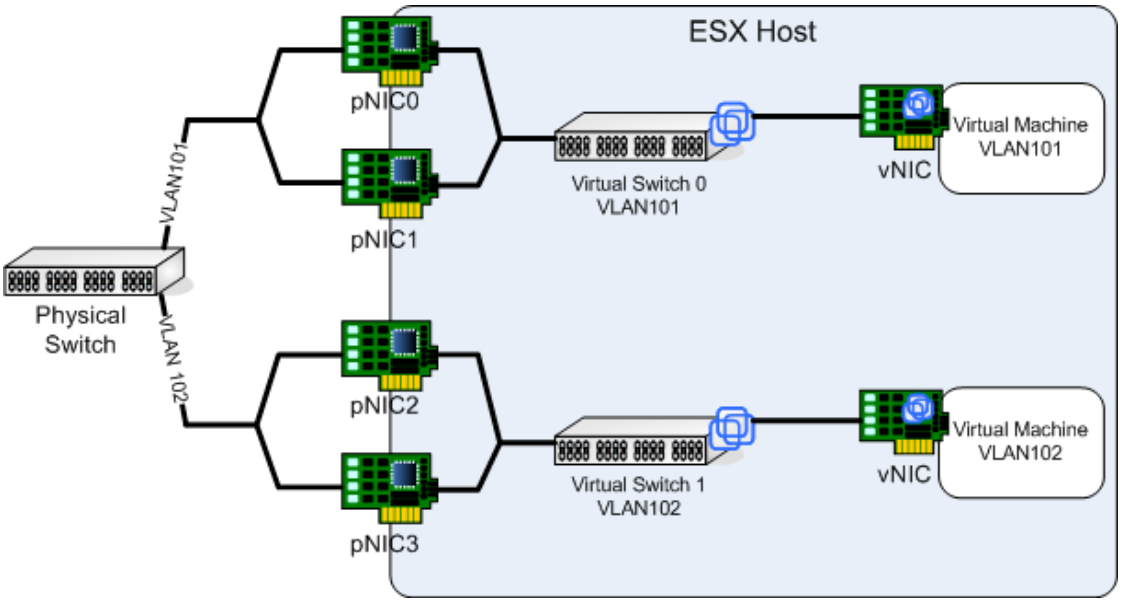
To help better integrate with the advanced nature of enterprise network architectures there are several ways to integrate VLAN configurations into a virtual infrastructure. There are three different ways that ESX allows us to configure VLANs within a host. Each method has a different impact on how our virtual switches are configured and how our guest operating systems interact with the network. In addition, there are advantages and drawbacks to each method.

External Switch Tagging (EST)

EST is the default configuration for all port groups within an ESX host. In this mode, all VLAN configurations are handled by the physical switch. This does not require any configuration on the physical switch other than properly configuring the ports that the physical NICs of the ESX host are plugged into for the proper VLAN. This configuration is no different than if a physical server were being plugged into the network that required a specific VLAN configuration.

In the following diagram, we have two virtual switches, each consisting of two Physical NICs. The physical switch ports that the Physical NICs are plugged into are configured for a specific VLAN. By design, ESX ensures that a particular VLAN is presented all the way to the virtual machine. The only way a virtual machine can communicate on the network is if it is assigned an IP address that falls within the subnet range that defines the specific VLAN that it is connected to.

Figure 6- 6: External Switch Tagging



Advantages to Using EST

- Easiest VLAN configuration (No additional configuration necessary)
- Supported in every version of VMware ESX.

Disadvantages to Using EST

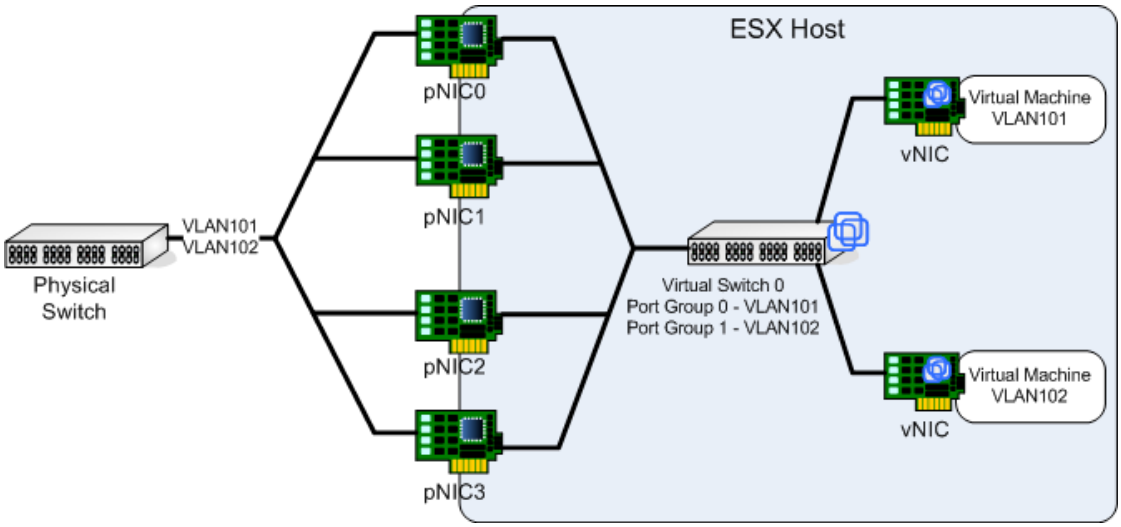
- Amount of VLANs that may be configured is limited by the number of physical NICs installed in the host
- Large number of NICs required for multiple VLAN configurations

Virtual Switch Tagging (VST)

VST consist of allowing a virtual switch to handle its own VLAN tagging. By configuring the uplink switch to explicitly tag packets with the 802.1q specifica-

tion, all control of VLAN configurations is handed over to the VMkernel. The processing of 802.1q tags is managed by the physical network adapter hardware, so overhead from these tags never hits the VMkernel and has no impact on virtualization processing. In this configuration, each physical switch port that connects to a physical NIC is configured in a trunk mode. After the trunk is established, specific VLANs are presented down the trunk to the uplink "switch" which, in the case of the diagram below, is an ESX virtual switch. A port group must then be created for each VLAN.

Figure 6- 7: Virtual Switch Tagging



Once the virtual switch receives the VLAN information from the trunk it needs to have the capability to assign virtual NICs to specific VLANs. In order to achieve this, port groups must be configured within ESX. Since a virtual switch does not have physical ports that we can assign specific VLANs to, we need to utilize port groups as a reference to a particular VLAN. Each VLAN that is being announced down the trunk to the virtual switch must have its own port group configured before virtual machines can be utilized on the published VLAN. It is best practice to set the Port Group Labels to "VLANX", where X is replaced by the published VLAN's ID.

You will need to review your switch vendor's documentation to ensure 802.1q VLAN tagging is a function of your specific switch model. Each physical switch also has different configuration steps for properly configuring 802.1q; so again, documentation should be consulted for the proper configuration. One thing to

note is that port groups cannot be configured with the "Native VLAN" for a switch. This Native VLAN is utilized for switch management and does not get tagged with a VLAN ID, and therefore is dropped by ESX. The default value for Cisco switches is 1, so any value between 2 and 4094 should work without issue.

Using VST also changes the way in which we recommend configuring virtual switches. One of the main reasons we recommend creating multiple virtual switches is so multiple VLANs can be utilized within a single ESX host. VST removes this restriction and allows a seemingly unlimited number of VLANs to be accessed through a single virtual switch. For this reason, if VST is to be utilized in an environment, we recommend configuring a single virtual switch that contains all Physical NICs in the system. This provides additional redundancy and throughput to the virtual switch, but simplifies the management of an ESX host by allowing us to configure port groups only once per ESX host.

Advantages of Using VST:

- A single virtual switch can utilize multiple VLANs, removing the dependency on multiple virtual switches and physical NICs to support multiple VLANs
- Once 802.1q trunks are established, adding new VLANs to ESX is a simple process
- Decreases the amount of network connections for ESX hosts that will support multiple VLANs

Disadvantages of using VST:

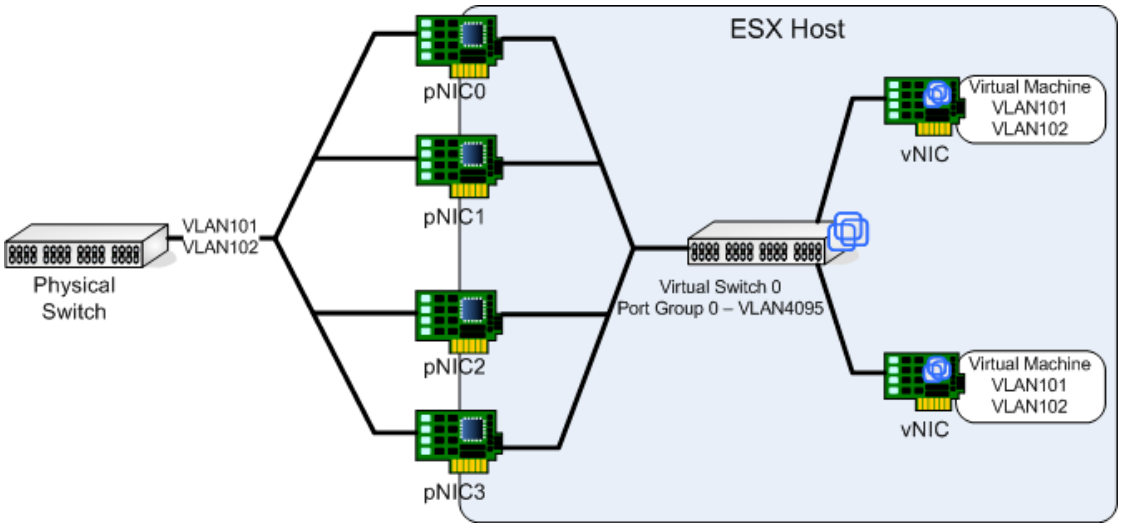
- May not be supported by all network infrastructures.
- Configuring trunk ports directly to servers is a new configuration in most environments, and is not always welcome.

Virtual Machine Guest Tagging (VGT)

The final mode for configuring VLANs for virtual machines is virtual guest tagging. In VGT mode, the virtual switch no longer reads 802.1q tags, but instead forwards them directly to the virtual machine. The guest operating system is then responsible for properly configuring the VLAN for the virtual NIC of the virtual machine. There is extremely limited support to this configuration. Most 2.4 and higher Linux kernels have VLAN tagging support built in. The only time this configuration should be utilized is if a particular virtual machine re-

quires access to more than 4 VLANs. This number is based on the fact that a single virtual machine may only utilize 4 virtual NICs.

Figure 6- 8: Virtual Guest Tagging



This configuration is achieved by configuring a new port group to use VLAN 4095. Any Virtual NIC assigned to this port group will then have VGT enabled.

Accessing the Infrastructure

After about 20 pages of reading about the key components it's about time we talked about how to actually communicate with your infrastructure. You will likely find that after reading this chapter up to this point that accessing the infrastructure is quite simple and anti climactic.

Service Console

Access to the service console is required for several specific reasons. First, VirtualCenter must communicate with its agent that is running within the service console operating system. This is how host and virtual machine configurations and performance metrics are captured and stored in the VirtualCenter database. Many users also find a need to install specific support applications such as hardware management or backup. This requires access to the service console to perform installations and configurations using standard RedHat Enterprise

Linux mechanisms. Others still require access to create custom automation scripts or harden the system to meet corporate IT security standards.

By default, ESX creates a Service Console port group on the host. This is represented within the service console as a device by the name of vswif0. Don't be thrown off by the name. To the service console, this interface acts identically to a typical eth0 interface of any Linux installation. ESX enables an iptables firewall on this interface to limit communication to the ESX host to only what is required for system management.

Without configuring firewall policies it is possible to connect to a host using SSH, HTTP, and through VMware's management protocol for VirtualCenter. It is possible to open additional access for items such as backup or monitoring agents by using either the CLI or through the Virtual Infrastructure Client. It is also possible to create multiple service console port groups if the console operating system is required to participate on multiple networks such as a production and a backup network.

We want to make the recommendation that a separate management network be configured for your Service Console and VMotion network connection. This will provide maximum security by setting up routing filters or potentially firewall policies to only allow specific communication to these sensitive components over certain ports or from certain workstations. Access to the service console should be treated with the highest level of security, as a compromise of the service console puts every virtual machine running on the host at risk of not only unauthorized power operations, but also potential data theft.

Virtual Machines

Assuming a virtual machine's virtual NIC is attached to public virtual switch your virtual machines will communicate with the network like any other device would. One of the most important things to note about virtual switches is that they do not have any layer 3 networking capabilities, so will not directly process packets destined for a different network. This type of communication must traverse the physical network and communicate through the physical network infrastructure. This provides a significant level of security around virtual machine communication.

There will be instances where two virtual machines are communicating with each other on the same port group of the same host. Since virtual switches operate at layer 2 of the network stack these virtual machines will communicate with each other without touching the physical network. It is important to remember this when performing troubleshooting of communication issues, as port monitoring at the physical switch may not show all traffic as expected.

In a private virtual switch virtual machines that reside within the same port group will be able to communicate with one another over the system bus due to the inner workings of the virtual switch. Virtual machines cannot communicate with one another in a private virtual switch unless there is a virtual machine that is configured on multiple port groups with IP forwarding available. This gateway virtual machine would need to be set as the default gateway for the virtual machines and will act as a router for inter communication. In a private virtual switch no communication is ever sent directly to the physical network.

Practical Configurations

With all of the options available for setting up a network infrastructure within ESX, many people are often overwhelmed when defining their network strategy. Except for the most extreme cases the best answer is often the easiest solution to implement. Just because VMware provides tons of functionality it doesn't mean you must try and integrate it into your design. It often just complicates the solution and negatively impacts performance overall, which is far from desired.

Standard ESX Installation Recommendations

We've already discussed some of the best practices around configuring your physical switches, physical NICs, virtual switches, and port groups. Now we want to take the opportunity to discuss the best way to use all of these components together in practical configurations that are commonly used across organizations of varying sizes.

Here we highlight a handful of the most practical use case scenarios for virtualization networking. There are cases that will require the use of more than four

network adapters such as providing internal and DMZ virtual machines that require physical network separation, but those are not as common as the configurations highlighted throughout the rest of the chapter.

3 NICs

The use of three physical network adapters was very common in ESX 2. Unlike with ESX 3, it was not easy to configure the service console in a redundant fashion, so there was little reason in configuring more NICs than could be used. As is the case now with ESX 3, having 2 physical gigabit adapters is more than enough to handle even the most intense ESX workloads.

Many systems today come with dual port on-board gigabit network connectivity. With a three NIC configuration an additional PCI gigabit adapter should be purchased and installed in the server. One port of the on-board adapter will be dedicated to the ESX service console and VMotion activity. The remaining on-board port and the PCI NIC port should be joined together to support networking capabilities for your virtual machines. This will provide redundancy for your virtual machines only, and, by default, will not provide any failover capability for your service console or VMotion connectivity.

The configuration of the virtual switches inside the ESX host itself will depend on the physical network layout of the infrastructure. There are two different methods of configuring this setup, which is dependent on the number of virtual switches leveraged.

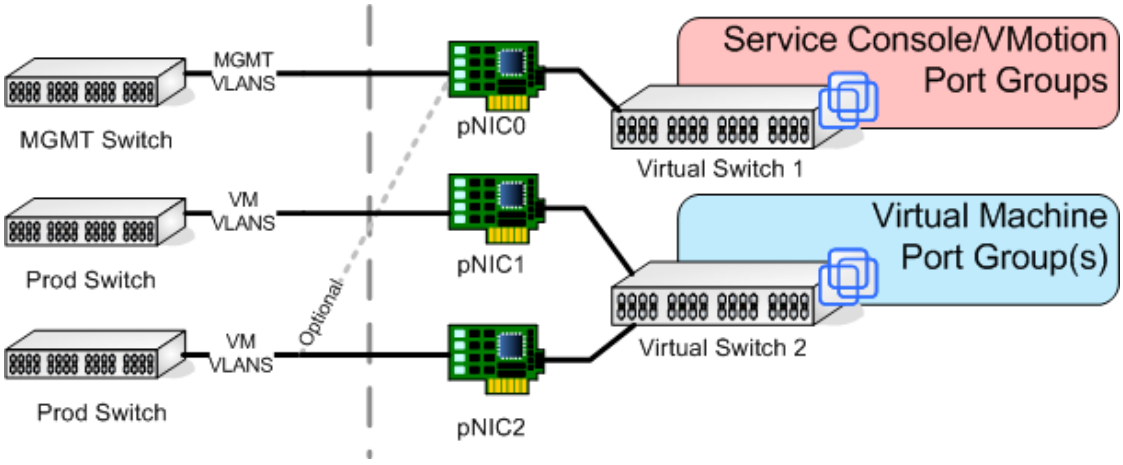
Two Virtual Switches

If the "management" network for the service console and VMotion connection use a different physical network infrastructure than the virtual machine network, two virtual switches will be required. In this case, it will also not be possible to provide any failover for the service console or VMotion, period. Configuring a failover NIC is only possible across port groups that reside on the same virtual switch.

The first virtual switch contains the first on-board networking port and is configured with two port groups; one for the service console and one for the VMkernel. If you do not plan on using VMotion, the latter port group is not

necessary. The second virtual switch is configured with the remaining on-board NIC port and the PCI NIC port. A virtual machine port group for each VLAN presented to the ESX host should be created. This will make up the virtual machine network environment and will have integrated redundancy due to the dual NIC switch configuration.

Figure 6- 9: 3 NIC/2 Virtual Switches



Advantages of 3 Physical NICs w/ 2 Virtual Switches

- Easy to set up and configure within ESX
- Provides physical isolation of management network and production network

Disadvantages of 3 Physical NICs w/ 2 Virtual Switches

- No redundancy for management functions such as service console access and VMotion

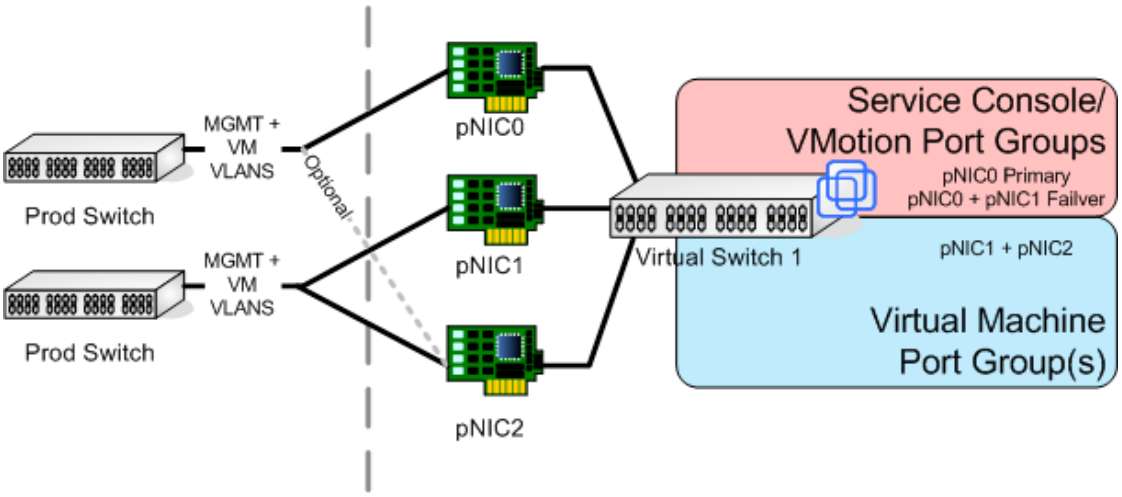
Single Virtual Switch

If the physical switches in the network will be providing all connectivity for management functions and virtual machine connectivity, it is possible to create a single virtual switch with at least three port groups: Service Console, VMkernel, and Virtual Machine (one per VLAN). The Service Console and VMkernel port groups should be assigned to the first on-board NIC port of the server with the two remaining NICs configured as failover adapters. The two remaining adapters should be added to a port group for virtual machine usage. These

adapters will automatically be configured in a failover mode based on the fact that they both belong to the same port group. You will also need to make sure that if you intend to provide failover for the management NIC that you also configure the physical switch ports that the virtual machine NICs are connected to by presenting the proper management VLANs down all paths. If this step is ignored a failed over service console/VMotion port group will have no way to communicate to the physical network.

The reason we want to assign specific NICs in this case is to ensure that management activity such as service console based backups and VMotion activity does not impact your virtual machine network access with the exception of a NIC/Port failure in the management port group. There would be no guaranteed separation if every port group in the virtual switch had access to every NIC as a primary adapter.

Figure 6- 10: 3 NIC/Single Virtual Switch



Advantages of 3 Physical NICs w/ 1 Virtual Switches

- Provides redundancy for management functions such as service console access and VMotion

Disadvantages of 3 Physical NICs w/ 1 Virtual Switches

- More complicated setup within ESX
- Potential production virtual machine impact during failover
- Not possible if Management Network is on separate physical switch infrastructure from Production Network

4 NICs

The most common configuration in an ESX 3 infrastructure is through the use of four physical NICs. Assuming your ESX host has come preconfigured with a dual-port on-board adapter you would simply add a second PCI dual-port gigabit adapter. The cost difference between a dual port and single port PCI adapter is actually quite minimal. Similarly to the various options available with a 3 NIC configuration, there are two different options when using a 4 NIC configuration based off your physical network design and/or how you wish to configure your virtual switches on the ESX host.

2 Virtual Switches

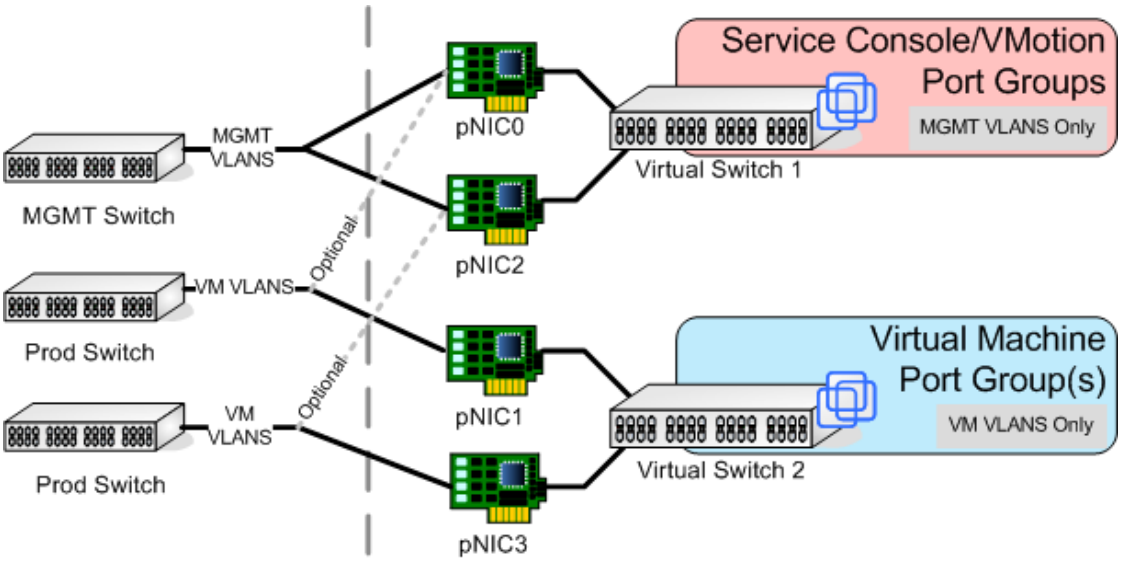
This is easily the most common of all configurations, and the one we most recommend during customer implementations. Whether you have a separate network infrastructure for management functions and production use, or you simply want to use two virtual switches, this design will provide the maximum redundancy, flexibility, and security for your virtual network infrastructure. Best of all, it's also very easy to setup and manage.

In this configuration two virtual switches are created, each consisting of two physical NICs. Each virtual switch should contain one connection to an on-board NIC port and one connection to a PCI adapter. This will ensure that even with a NIC failure, the virtual switch will maintain connectivity. On the physical network side, each on-board NIC port should connect to the same physical switch. The two PCI NIC ports should also both connect to the same physical switch, but a different physical switch than the on-board ports if possible. This provides protection in the event a switch fails. In this configuration it would take a very significant disaster to bring the network of the virtual infrastructure down.

The first virtual switch will only contain your virtual machine port groups required for virtual machine access. The second virtual switch will only contain the port groups necessary for management such as for the service console and VMotion. Each virtual switch has full redundancy based on the fact that each has two physical NICs, which in turn are optimally connected to alternate physical switches in the network infrastructure. This configuration also provides maximum security to the infrastructure by separating management and

production traffic. It is difficult in this configuration to accidentally put a virtual machine on the management network and vice versa.

Figure 6- 11: 4 NIC / 2 Virtual Switch



Advantages of 4 Physical NICs w/ 2 Virtual Switches

- Provides maximum redundancy for both virtual machine access and management tasks
- Most secure network configuration for your ESX environment by providing isolation of your production and management networks
- Simple setup and Easy to setup and configure within ESX

Disadvantages of 4 Physical NICs w/ 2 Virtual Switches

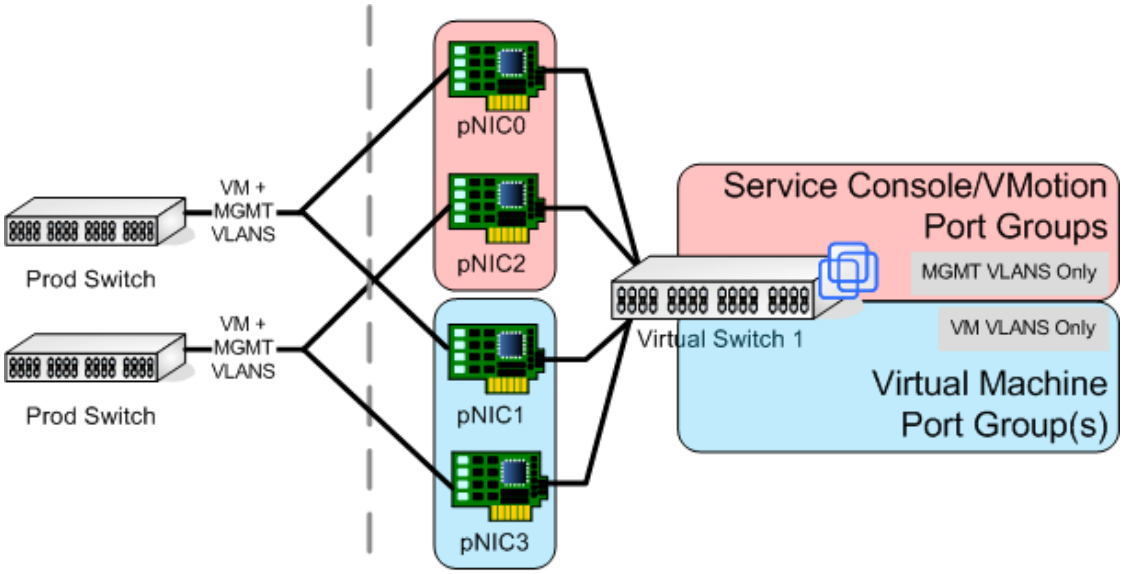
- There are no obvious disadvantages to this configuration

Single Virtual Switch

A 4 NIC Single virtual switch implementation is almost as simple to configure as the previous 4 NIC design. The primary difference between the two designs is in the fact that all VLANs for the entire solution are presented down all network paths, eliminating physical separation of your management and virtual machine networks.

By default, this solution will assign traffic for all port groups down all four paths of the host. There is potential for a spike in traffic such as from a virtual machine backup or a VMotion migration to impact the performance of the configured virtual machines. To prevent this, we recommend that if you are to use this solution, that you still configure two Physical NICs (Each connected to a different physical switch) as primary for your virtual machine port group(s), with the remaining two NICs (also connected to separate physical switches) as failover NICs. The two NICs that were configured for failover for the virtual machine port group(s) should be configured as the primary for the management port group(s), with their failover being the primary NICs from the first port group. A completely unnecessary amount of micromanagement goes into properly configuring this scenario so we recommend that you simply follow our previous recommendation and use two virtual switches.

Figure 6- 12: 4 NIC / 1 Virtual Switch



Advantages of 4 Physical NICs w/ 1 Virtual Switches

- Provides maximum redundancy for both virtual machine access and management tasks

Disadvantages of 4 Physical NICs w/ 1 Virtual Switches

- Micromanagement of Active/Passive NICs per port group if properly implemented

- No physical separation of management and virtual machine networks
- Not possible if Management Network is on separate physical switch infrastructure from Production Network

Worst Case Scenarios

There are always going to be cases in which following the best practices for a solution will not be possible. Whether it is due to hardware limitations or simple corporate policy, there are some configurations that will need to be considered for some organizations. For that reason we would like to point to the two of the most common "worst case scenarios" that we have come across.

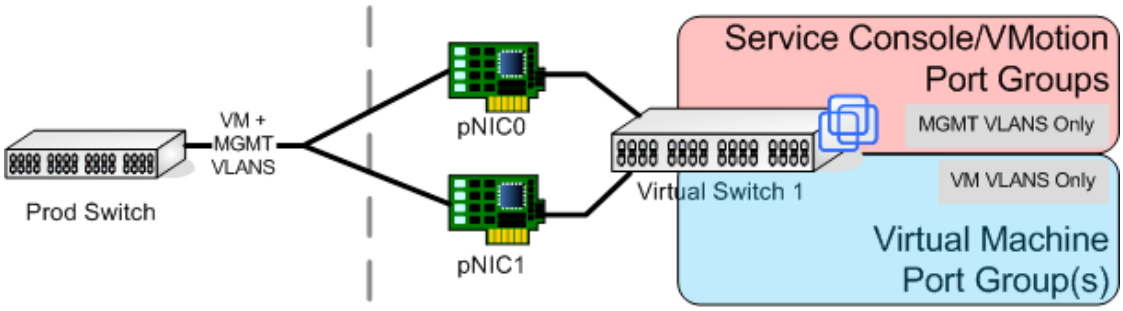
Limited to 2 Physical NICs

We are hoping that this situation is limited to the development space, as we really hate to see someone not able to spring the extra $125 and put in a second dual port adapter into an ESX server for production use. There is nothing wrong with this scenario for development, but we need to set a few expectations.

In this situation, if at all possible, plug the two NICs into different physical switches. This will at least protect the environment in the event one of the switches was to fail. For a development environment, there is no shame in plugging both NICs into a single switch. To keep redundancy and load balancing for the virtual machines a single virtual switch configured with both physical NICs should be configured. This virtual switch will be configured with all the required port groups for virtual machine and management use.

We will forgo our traditional Advantages/Disadvantages for this configuration, since there really is nothing but disadvantages here. VMotion and virtual machine backups will potentially interfere with your running virtual machines and there is no physical separation of your management and virtual machine networks (assuming they are different networks at all in this situation).

Figure 6- 13: 2 Physical NICs



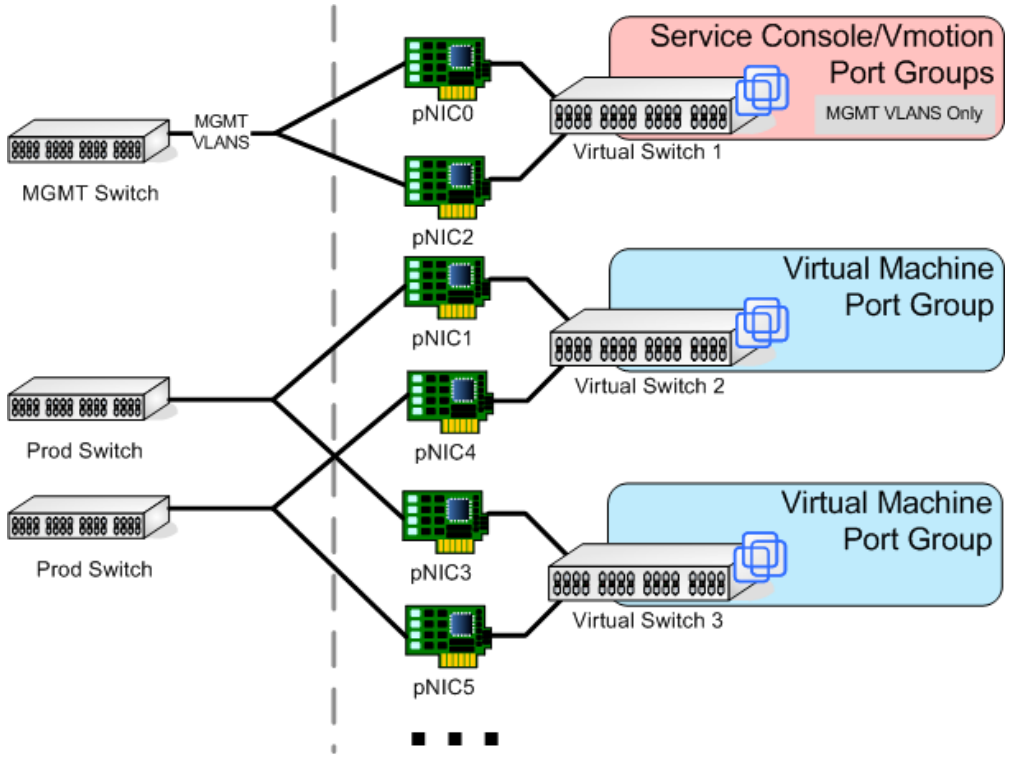
VLAN Tagging Not Allowed

There is no technical reasoning that a network engineer can conceive that should prevent the use of VLAN tagging on your ESX host (Unless of course your switches don't support it). It is strictly a political battle and should be dealt with organizationally. For the cases where a fight to the death is not possible and you have no choice but to not enable VLAN tagging for your ESX hosts there is still a solution that will technically work, but it is not glamorous.

In this scenario you will need at least one, and preferably two NICs for every VLAN that you wish to configure your virtual machines on. Each NIC or pair of NICs will need to be configured in a virtual switch with a single virtual machine port group configured. The network hardware and connectivity adds up very quickly if you have to provide connectivity to four, five, or even eight different VLANs on every ESX host in the environment.

Again, if there is any way to avoid this situation, please do. It will make your experiences with virtualization that much more successful and enjoyable.

Figure 6- 14: No VLAN Tagging



As you can see, there are many options available for properly configuring your network environment for your virtual infrastructure. We've provided the most common configurations found in virtual infrastructures of various sizes. By leveraging our examples and considering the advanced configurations discussed throughout this chapter, we are confident you now have all of the information necessary to provide a solid network design that meets the needs of your organization.