VMware Virtual Network Concepts - Summarized (VI3)

This document summarizes virtual networking (denoted as vNetworking this point on) concepts as described in whitepaper titled, "VMware Virtual Networking Concept". Complete document is available at http://www.vmware.com/files/pdf/virtual_networking_concepts.pdf.

Throughout this document we will be using certain terms that are only relevant to this document and are used to distinguish between Physical Networking (denoted as pNetworking this point on) concepts and vNetworking concepts. These are:

- vNetwork – Virtual Network
- pNetwork – Physical Network
- vNIC – Virtual NIC (Network Interface Card)
- pNIC – Physical NIC
- vSwitch – Virtual Switch
- pSwitch – Physical Switch
- vPort – Virtual Port on vSwitch
- pPort – Physical Port on pSwitch
- vUplink – Same as uplink in VMware terminology. This describes a connection from a vPort to pNIC in an ESX host.
- pUplink – Physical connection from pNIC on an ESX host to a pPort on a pSwitch.
- vVLAN – Tagging at vNetwork devices (i.e. vSwitch, vNIC, etc.)
- pVLAN – Same as VLAN in pNetworks
- vTrunk – Tagging at a vSwitch Level
- pTrunk – Same as Trunk in pNetworks
- PG – Port Group as defined in VMware terminology
- ISL – Interswitch Link as defined in pNetworking
- DTP – Dynamic Trunking Protocol as described in pNetworks
- MTU – Maximum Transmission Unit as described in pNetworks
- STP – Spanning Tree Protocol as defined in pNetworks
- vStack – VM TCP/IP Networking stack
- VIC – Virtual Infrastructure Client

vNIC Types

1. **vlance** – Uses AMD 79C970 PCnet32 LANCE vNIC (10 Mbps) Emulation driver. Does NOT require VMware Tools except Vista and later versions.
2. **e1000** – Uses Intel 82545EM Gigabit Ethernet vNIC Emulation driver (64-bit/32-bit). Does NOT require VMware Tools.
3. **vmxnet** – Strictly vNIC that's optimized for performance. MUST INSTALL VMware Tools.

4. **vmxnet2** – Similar to vmxnet but provides enhanced features such as jumbo frames, hardware offloads, etc.
5. **vmxnet3** – Combined functionality of vmxnet and vmxnet2 plus ipv6 offloads, msi/msi-x interrupt delivery, multiqueue support, etc.
6. **Flexible** – Used by VMs most often. Its type is **vlance** on boot and **vmxnet** once VMware Tools are installed.
7. **vswif** – Similar to vmxnet but used by SC
8. **vmknic** – vNIC in VMkernel used to manage pNICs on a host. It's used for TCP/IP stack, VMotion, NFS, Software iSCSI connections, remote console

NOTE: Speed & Duplex settings are irrelevant in VMs because everything is done in RAM.

vSwitch

A virtual switch is simply a core L2 forwarding engine that does VLAN tagging, stripping, filtering, L2 security, checksum, segmentation offload units, and many other tasks that are done by pSwitches in pNetworks. It is both similar and different from pSwitches.

Similarities:

- CAM Tables (Not a Hub)
- Access Ports via vSwitch Tagging
- Trunking via Guest Tagging
- Mirror Ports via Promiscuous mode

Differences:

- No IGMP snooping or Unicast needed for multicasting. This is very useful when doing NLB.
- STP not needed (Except when bridging is used in guest)
- Cannot connect different vSwitches to same pNIC on a host

VLAN (802.1q) Modes:

1. VST – Virtual Switch Tagging – Most commonly used and have following features
    - one PG per VLAN
    - VM assigned to a PG (not vSwitch)
    - vSwitch removes tags for all outbound and in inbound frames
    - pTurnking is required for multiple VLAN traffic
    - *VST mode does not support DTP so you have to make the trunk static and unconditional*

2. VGT – Virtual Guest Tagging
    - 802.1q tagging is done inside a VM using an 802.1q tagging engine.
    - vSwitch preserves the tags between vStack in VM and pSwitch

- pTurnking is required for multiple VLAN traffic

3. EST – External Switch Tagging – Tagging is done at pSwitch.
    - pTrunking is necessary
    - Limited to number of pNICs available in a host when port-based VLANs are used

Use of Native VLAN (VLAN 1): VMware does NOT recommend the use of VLAN 1 in virtual environment. In the event that you have to associate VLAN 1 with a PG and pass VM network traffic through it, you must do one of the following two things:

- Make sure VLAN 1 is not the native VLAN on pSwitches. You may change the default native VLAN to another VLAN ID.
- Enable the native VLAN 802.1Q tagging capability. Some pSwitches do not support this option and some other pSwitches do not need it as tagging on the native VLAN is enabled by default.

Note that when you change the behavior of the native VLAN on one of your external pSwitches, by doing either step above, you will likely need to change all the neighbor pSwitches as well so they can still communicate on the native VLAN properly.

L2 Security Options:

- Promiscuous: Prevents other VMs on the same switch from seeing unicast traffic to other nodes
- MAC Lockdown: Prevents VM from changing its own Unicast address.
- Froged Transmit Blocking: Prevents VMs from sending traffic that appears to come from nodes on the network other than themselves

PGs

- Port Group is basically a label with many common characteristics used by multiple vPorts such as
    - vSwitch name
    - VLAN Ids
    - Tagging & Filtering Policies
    - Teaming Policy
    - L2 Security option
    - Traffic Shaping
- PGs are very important for VMotion.
- PGs are similar to **smartport** option found on Cisco Switches.

vUplink (uplink)

An uplink or vUplink as denoted here in this document is a virtual connection from vPort on a vSwitch to a pNIC in a host.

For multiple VLAN traffic to pass through a vUplink it is important that pPorts connected to pNICs on a host must be in a pTrunk on the pSwitch.

NIC Teaming

- Teaming is configured at the vSwitch level and policies can be applied both at the vSwitch level and PG level. However, PG teaming policies will override vSwitch teaming policies.
- pPorts on a pSwitch in the same team form the same L2 Broadcast domain just like VLANs.
- **Teaming State:**
    - Maintained for each PG
    - pNIC failures are transparent to vNICs
- **Load Balancing Types & Policies:**
    - *<u>Route based on the originating vPort:</u>*
        - Chooses vUplink based on which vPort traffic entered in the vSwitch (Attached to same pNIC) unless there is a pNIC failure in the team.
        - Replies are received on the same pNIC as the pSwitch learns port association
        - Equal Distribution if vNICs are greater than pNICs
        - A VM cannot use more than 1 pNIC at a given time unless it has more than one vNIC
        - *Slightly less load than MAC based hashing*
        - *Don't use port-channel or bonding on pSwitch if using srcPortID (Route based on the originating virtual switch port ID) or srcMAC (Route based on source MAC hash)*
    - *<u>Route based on source MAC hash:</u>*
        - Chooses vUplink based on source MAC
        - Replies are received on the same pNIC as the pSwitch learns port association
        - A VM cannot use more than one pNIC at a time unless it uses multiple MAC addresses for traffic it sends (i.e. NLB in multicast mode)
        - *Don't use port-channel or bonding on pSwitch if using srcPortID (Route based on the originating virtual switch port ID) or srcMAC (Route based on source MAC hash)*
    - *<u>Route based on IP hash:</u>*
        - Chooses vUplink based on the source and destination IP addresses
        - Equal distribution depends on the number of TCP/IP sessions to unique destinations
        - Can use Link Aggregation

        - *pSwitch sees vNIC MAC address(s) on multiple pPorts. There is no way to predict which pNIC will receive inbound traffic*. (Must be load-balanced for inbound traffic at pSwitch)
        - *All pNICs must be attached to same pSwitch or pSwitch stack (802.3ad compliant) and configured to use link-aggregation in static mode (no LACP)*
        - All pNICs must be active
        - All PGs must inherit settings from vSwitch that they are on.
- **Failover Configurations:**
    - *Link Status only:*
        - Status provided solely by pNICs (physical failures only)
    - *Beacon Probing:*
        - Uses Link Status and Beacon Probes (Broadcast Frames) sent/received by pNICs in a team to detect upstream failures.
        - In case of a failure, vSwitch reroutes traffic to use a different pNIC in the team.
    - *Failback Policy:*
        - By default Failback is enabled meaning "Rolling Failover" in VIC is set to NO. When primary pNIC is experiencing intermittent connections, disable STP on pPorts connected to pNICs on ESX host or use port-fast features found on some Cisco pSwitches. This saves about 30 seconds during pSwitch initialization, Etherchannel, PAgp/LACP must be disabled because they are not supported, Trunking negotiations saves about 4 seconds set "Rolling Failover" to YES in VIC.
    - *Failover Order Policy:*
        - Distribute work load for pNICs on the host
    - *Notify Switches Policy:*
        - When failover occurs, ESX notifies pSwitches
        - When YES is selected in VIC, vNIC traffic is rerouted by vSwitch to a different vUplink and CAM is updated on by pSwitch
        - When NO is selected in VIC, vNIC traffic is not rerouted. This is useful when NLB is configured in unicast mode.

VMware Recommendations:

- *For best security, VMware recommends you use a dedicated virtual switch or, at a minimum, a dedicated virtual switch VLAN port group for VMotion*
- DO NOT use VLAN 1 as your native VLAN.
-