Why This Talk?

- A vast majority of networking problems are configuration issues with the physical switch.

- Physical switches are managed by network administrators. Virtual switches are under the control of ESX administrators.

- Enabling/disabling various networking features can have subtle or drastic implications on your network connectivity.

- Knowledge on how virtual switch works helps to troubleshoot problems.
Outline

- ESX Networking details
- Scenarios
  - Virtual Switch boundaries
  - VLAN
  - Layer 2 Security
  - Load Balancing
  - Failover
- Diagnostics

This talk assumes familiarity with ESX networking features
ESX Networking: Logical Layout

- Multiple layers, multiple ways to interconnect
- Interesting possibilities!

---

Virtual Machine

Virtual NIC

Virtual Switch

Physical NIC

ESX

Physical Switch

Physical Machine
ESX 3.x Networking Details

Service Console
- user
  - kernel
    - TCP/IP stack
      - netcos driver
  - vmninmod

VM
- user
  - kernel
  - TCP/IP stack
  - vlance / vmxnet / e1000
  - Net Device emulation
  - VMX
  - SCSI emulation

VMKernel
- COS client
- vmxnet client

Portgroup
- Ethernet Switching
- VLAN Tagging
- L2 Security
- Load Balancing
- Failover
- ... (Virtual Switch)

Virtual Switch
- vmklinux compatibility layer
- Linux device drivers

All the Physical NICs are managed by VMkernel.

VMKernel uses its own TCP/IP stack.

Clients inherit the properties of portgroup.

Portgroup policies define what features are used.

New functionalities could be plugged.

All network traffic has to go through some virtual switch.

VMKernel uses modified Linux drivers.

1: VI traffic to COS
2: VM traffic
3: VI traffic for Remote Console
4: IP storage
5: VMotion traffic
Virtual Switch

- Operates at Layer 2, no layer 3 functionalities.
- Can have zero or more uplinks (Physical NICs)
- Cannot share (uplinks) physical NICs with other virtual switches
- To use a virtual switch there should be at least one portgroup defined
Portgroups do not segment broadcast domain

- VLANs segment broadcast domains

Clients inherit the properties of the portgroups (in ESX 2.x properties are specified to the virtual NIC)

- Portgroup policies Overrides virtual switch policies.

Can use subset of NICs available to the virtual switch

- Can share NICs with other portgroups on the same virtual switch

Implication: Same set of Physical NICs can be used with different policy settings. For ex. VLAN, NIC teaming etc.
Virtual Switch: External View

- Virtual Switch behaves like a dumb switch
- Does not speak
  - STP - Don’t have to, No Loops possible
- Does not speak DTP, VTP, ISL etc
- Does not speak LACP
  - Physical Switch ports have to be aggregated in Manual mode
- Optional CDP support planned for the future version

Multiple Client MAC addresses appear on this port
Virtual Switch: Internal View

- MAC address learning
  - Unlike physical switches Virtual Switch does not learn MAC addresses from the traffic flow
  - Virtual NICs notify MAC address when they register
  - Every other unicast MAC address belong to uplink port

- Link negotiation
  - Virtual NIC does not negotiate speed/duplex with the virtual switch
  - Virtual NICs do not reflect the speed/duplex state of the Uplink (physical NIC)
  - Guest reports link down status when the virtual ethernet device is disconnected in the UI
Virtual Switch Boundaries

- Virtual switches are isolated. i.e. Trunking is not possible between virtual switches. Only uplinks connect virtual switches.
- Communication from VM A to VM B can happen only through external network
Virtual Switch Boundaries

- Virtual Machines can interconnect Virtual Switch
- Virtual NICs need to be placed in different subnet to use both virtual switches
- Layer 2 Loops possible if the VM acts like a bridge
Virtual Switch Boundaries

- VMKernel TCP/IP Stack routing table determines packet flow
- Put IP Storage and VMotion on separate subnets for isolation
- Traffic will go through the same virtual switch if they are in the same subnet
VLAN: Why Trunk?

- Uplink in a virtual switch is a trunk link to the physical switch.
- Configure the physical switch port as a trunk port to allow traffic with tagged frames.
Native VLAN

- Physical Switch does not tag frames on the Native VLAN
- Virtual Switch does not have the notion of Native VLAN
- Communication A – B fails: Virtual switch forwards only tagged frames to B
- Communication B – A may or may not fail: Physical switch may or may not accept tagged frames on native VLAN
- Workaround: Put VM B on an portgroup with no VLAN tagging or enforce tagging on switch port P2
Virtual Switch VLAN Behavior Example

- Loopback cable interconnects VLAN 5 and VLAN 10 into the same broadcast domain
- VM A and VM B can talk to each other
- In ESX 2.x the response packets from VM B will not reach VM A. Path optimization prevents this communication
- ESX 3.x avoids this problem
VLAN id 4095 enables VGT mode in ESX 3.x.

- In VGT mode guest can send/receive any VLAN tagged frame (0-4094).
- Virtual switch does not filters VLAN.
- Filtering could be done in the physical switch port.
- However VM B could still talk to VM A.

Filter: Allow VLAN: 7, 10
Layer 2 Security

- ESX Layer 2 security options give a level of control beyond what is usually possible in physical environments

- **Promiscuous Mode: Deny**
  - Virtual NIC will appear to go into promiscuous mode, but it won’t receive any additional frames

- **Forged transmits: Deny**
  - drop any frames which the guest sends with a source MAC different from the one currently registered

- **MAC address changes: Deny**
  - if the guest attempts to change the MAC address to something other than what’s configured for the virtual HW, stop giving it frames
Layer 2 Security

- Why “Deny MAC Address Changes”?
  - Guest can change its MAC address to send spoofed frames
  - Guest can change its MAC address to listen to other traffic when promiscuous mode is denied.

- To restrict the VM to use only its MAC address enforce “Deny MAC Address Changes” and “Deny Disallow Forged transmits”

- Deny all three options for complete layer 2 security
Layer 2 Security: Interactions

- Microsoft Network Load Balancing
  - **Deny Forged transmits** will break Microsoft Network Load Balancing operating in Unicast mode
  - In Unicast mode Cluster nodes use fake MAC address for outgoing traffic to prevent switches from learning true MAC address. This technique allows the incoming traffic for the cluster IP to be sent to all the ports of the physical switch.
Layer 2 Security: Interactions

- Windows IP address conflicts
  - **Deny Forged transmits** will cause machines on the network to point to the offending machine instead of defending machine in the case of IP address conflict.
  - Windows Sends gratuitous ARP (ARP request for its own IP) to detect duplicate IP address. If a host responds back, then duplicate IP.
  - In the event a host responds back (duplicate IP found), windows sends forged ARP request containing the MAC address of the defending machine. This updates the ARP table of the machines in the network with the IP address of the defending machine.
Switch Notification

- Client MAC address is notified to the physical switch using RARP frame
- When?
  - Whenever Client register itself with virtual switch
  - VM power on, Vmotion, Changing MAC, Teaming status change etc
- Why?
  - Allows the physical switch to learn MAC immediately
- Why RARP ?:
  - L2 broadcast reaches every switch
  - Doesn’t disrupts ARP cache
  - L3 information not needed to send RARP
Switch Notification: VMotion

- VMotion moves the VM from one switch port to another.
- Virtual Switches on source and destination should have identical L2 security policy (VC Checks this).
- Source and destination port should be in the same broadcast domain (implies same VLAN).
- Virtual NIC is unplugged on the source and plugged back at the destination host – triggers switch notification.
Load Balancing: Source MAC/Originating Port ID

- Outbound NIC is chosen based on source MAC or originating port ID
- Client traffic is consistently sent to the same physical NIC until there is a failover
- Replies are received on the same NIC as the physical switch learns the MAC/switch port association
- Better scaling if: no of vNICs > no of pNICs
- VM cannot use more than one Physical NIC unless it has two or more virtual NICs
Load Balancing: IP Hash (out-IP)

- Outbound NIC is chosen based on “Source-destination L3 address pair”
- Scalability is dependent on the no of TCP/IP sessions to unique destinations. No benefit for bulk transfer between hosts
- Physical switch will see the client MAC on multiple ports
  - Can disrupt MAC address learning on the physical switch
  - Inbound traffic is unpredictable.
NIC Teaming: Packet Reflections

- Broadcast / Multicast packets return to the VM through other NICs in the team
- Most Guest OS’es ignore duplicate packets
- Avoid NIC Teaming if the VM relies on frequent broadcast / multicast packets (for ex. Microsoft Network Load Balancing)
- ESX 3.x filters packet reflections
  - Frames received on wrong link is
    - Discarded in source mac/originating port id mode
    - Allowed in out-ip mode
Link Aggregation

- Allows load balancing of incoming traffic.
- Packet reflections are prevented - Aggregated ports do not re-send broadcast / multicast traffic
- Works well with out-ip since aggregated ports share a single entry in the MAC lookup table
- Throughput aggregation benefits are less relevant with the advent of gigabit and 10G Links
- Traffic flow is unpredictable
- Source mac/Source port id mode load is incompatible with Link aggregation in ESX 3.x
NIC Teaming: Multi Switch Configuration

- Physical NICs can be connected to different switches as long as they remain in the same broadcast domain
- Physical switches should be trunked or ISL’ed
- Expect problems if the port on each physical switch is configured with different VLAN/trunking options
- IP-hash (out-ip) mode is not recommended
  - Client MAC address can appear on all the physical switches
  - Client MAC address can appear on trunk ports
**NIC Teaming: Multi Switch With Link Aggregation**

- Same scenario as before, but uses link aggregation on each switch.
- Currently ports from different physical switches could not be aggregated into a single link.

**Diagram Description:**
- Virtual Switch
- Physical Switches
- Broadcast domain
NIC Teaming: Failover Scenarios

- Failover detection
  - Ethernet Link failure
  - Switch failure (beaconing)
- Fail-back
  - Rolling failover: No - Fail back is on
- Failover order
  - Order of Standby Adapters
- Unused Adapters – NICs excluded from teaming
- Changing the Order of Active Adapters switches the traffic flow through the NICs
NIC Teaming: Failover Implications

- Fail-back is on by default. If link is flaky physical switch will notice client MAC address on multiple ports frequently.
- Virtual switch uses the link as soon as it is up. Physical switch port may not accept traffic immediately when the link comes online.
- To minimize delays disable:
  - STP (use portfast mode instead) – 30 secs
  - Etherchannel negotiation, like PAgP (use manual mode) – 15 secs
  - Trunking negotiation – 4 secs
  - Link autonegotiation (Speed/duplex settings) – 2 secs
## Diagnostics: Link state

```
[root@mojave root]# esxcfg-nics -1
Name   PCI    Driver   Link  Speed    Duplex   Description                      
vmnic1 03:0d.00 e1000   Up  1000Mbps Full Intel Corporation PRO/1000 MT Server Adapter
vmnic0 04:0b.00 3c90c   Up  100Mbps  Full  3Com Corporation 3c905C-TX/TX-M [Tornado]
vmnic2 04:0e.00 e100   Down 0Mbps  Half Intel Corporation EtherExpress PRO/100 S Desktop Adapter
```

### Hardware
- **Processors**
- **Memory**
- **Storage (SCSI, SAN, and NFS)**
- **Networking**
- **Storage Adapters**
  - **Network Adapters**

### Network Adapters

<table>
<thead>
<tr>
<th>Device</th>
<th>Speed</th>
<th>Configured</th>
<th>vSwitch</th>
<th>Networks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PRO/1000 MT Server Adapter</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vnic1</td>
<td>1000</td>
<td>Full</td>
<td></td>
<td>vSwitch1</td>
</tr>
<tr>
<td><strong>3c905C-TX/TX-M [Tornado]</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vmnic0</td>
<td>100</td>
<td>Full</td>
<td></td>
<td>vSwitch0</td>
</tr>
<tr>
<td><strong>EtherExpress PRO/100 S Desktop Adapter</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vnic2</td>
<td></td>
<td>down</td>
<td></td>
<td>10.17.213.1-10.17.213.254</td>
</tr>
</tbody>
</table>

© 2006 VMworld
Diagnostics: Portgroup settings

```
[root@mojave root]# esxcfg-vswitch -l
Switch Name   Num Ports Used Ports Configured Ports Uplinks
vSwitch0      32       4          32           vnic0
  PortGroup Name           Internal ID VLAN ID Used Ports Uplinks
  NFS                      portgroup6   0      1       vnic0
  Service Console          portgroup0   0      1       vnic0

Switch Name   Num Ports Used Ports Configured Ports Uplinks
vSwitch1      64       3          64          vnic1
  PortGroup Name           Internal ID VLAN ID Used Ports Uplinks
  VMkernel                 portgroup3   0      1       vnic1
```

Networking

Virtual Switch: vSwitch0
- Service Console Port
  Service Console
  vsw0: 10.17.213.182
  Physical Adapters
  - vmnic0
  100 Full

Virtual Switch: vSwitch1
- VMkernel Port
  VMkernel
  10.2.0.50
  Physical Adapters
  - vmnic1
  1000 Full
Diagnoses: VMKernel TCP/IP Stats

```bash
> cat /proc/vmware/net/tcpip/ifconfig
```

```
[root@mojave net]# cat /proc/vmware/net/tcpip/ifconfig
Usage: plumb <portSetName> <ipAddress> [netmask]
Usage: unplumb <portSetName>
Usage: gateway <gatewayAddress>

<table>
<thead>
<tr>
<th>Name</th>
<th>Port</th>
<th>Address</th>
<th>Netmask</th>
</tr>
</thead>
<tbody>
<tr>
<td>vmk0</td>
<td>portgroup3</td>
<td>10.2.0.50</td>
<td>255.255.0.0</td>
</tr>
<tr>
<td>vmk3</td>
<td>portgroups</td>
<td>10.17.213.197</td>
<td>255.255.255.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Mtu/TSO</th>
<th>Network</th>
<th>Address</th>
<th>Ipkts Ierrs</th>
<th>Ibytes</th>
<th>Opkts Oerrs</th>
<th>Obytes</th>
<th>Coll</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>i0</td>
<td>16384/0</td>
<td>&lt;link#1&gt;</td>
<td>0 0 0 0 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i0</td>
<td>16384/0</td>
<td>127</td>
<td>127.0.0.1</td>
<td>0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vmk0</td>
<td>1500 /0</td>
<td>&lt;link#2&gt;</td>
<td>00:50:56:6e:49:2b</td>
<td>516985 0 419117210 470879 0 421234014 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vmk0</td>
<td>1500 /0</td>
<td>10.2/16</td>
<td>10.2.0.50</td>
<td>516985 0 419117210 470879 0 421234014 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vmk3</td>
<td>1500 /0</td>
<td>&lt;link#3&gt;</td>
<td>00:50:56:65:d5:21</td>
<td>1456953 0 187423828 829133 0 1873803352 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vmk3</td>
<td>1500 /0</td>
<td>10.17.213/24</td>
<td>10.17.213.197</td>
<td>1456953 0 187423828 829133 0 1873803352 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Routing:
- 0 bad routing redirects
- 0 dynamically created routes
- 0 new gateways due to redirects
- 56 destinations found unreachable
- 0 uses of a wildcard route

Routing tables:

Internet:

<table>
<thead>
<tr>
<th>Destination</th>
<th>Gateway</th>
<th>Flags</th>
<th>Refs</th>
<th>Use</th>
<th>Netif</th>
<th>Expire</th>
</tr>
</thead>
<tbody>
<tr>
<td>default</td>
<td>10.17.213.253</td>
<td>Ucc</td>
<td>0</td>
<td>799</td>
<td>vmk3</td>
<td></td>
</tr>
<tr>
<td>10.2/16</td>
<td>link#2</td>
<td>UC</td>
<td>0</td>
<td>0</td>
<td>vmk0</td>
<td></td>
</tr>
<tr>
<td>10.17.213/24</td>
<td>link#3</td>
<td>UC</td>
<td>0</td>
<td>0</td>
<td>vmk3</td>
<td></td>
</tr>
<tr>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>UH</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>
```


Diagnostics: vmkping

- ping command uses service console TCP/IP Stack
- vmkping uses VMKernel TCP/IP stack
Diagnostics: Collecting Network Traces

- Run tcpdump/ethereal/netmon inside the guest or in the service console
- Traffic visibility depends on the portgroup policy settings
  - Portgroup with VLAN id 0 (No VLAN)
    - Sees all the traffic on the virtual switch without VLAN tags
  - Portgroup with VLAN id ‘X’ (1-4094)
    - Sees all the traffic on the virtual switch with VLAN id ‘X’
  - Portgroup with VLAN id 4095
    - Sees all traffic on the virtual switch
    - Traffic is captured with VLAN tags
- Promiscuous mode
  - Accept: All visible traffic
  - Reject: Only traffic matching the client MAC address
Please remember to complete your **session evaluation form** and return it to the room monitors as you exit the session.

The presentation for this session can be downloaded at [http://www.vmware.com/vmtn/vmworld/sessions/](http://www.vmware.com/vmtn/vmworld/sessions/)

Enter the following to download (case-sensitive):

- **Username:** cbv_rep
- **Password:** cbvfor9v9r
Some or all of the features in this document may be representative of feature areas under development. Feature commitments must not be included in contracts, purchase orders, or sales agreements of any kind. Technical feasibility and market demand will affect final delivery.