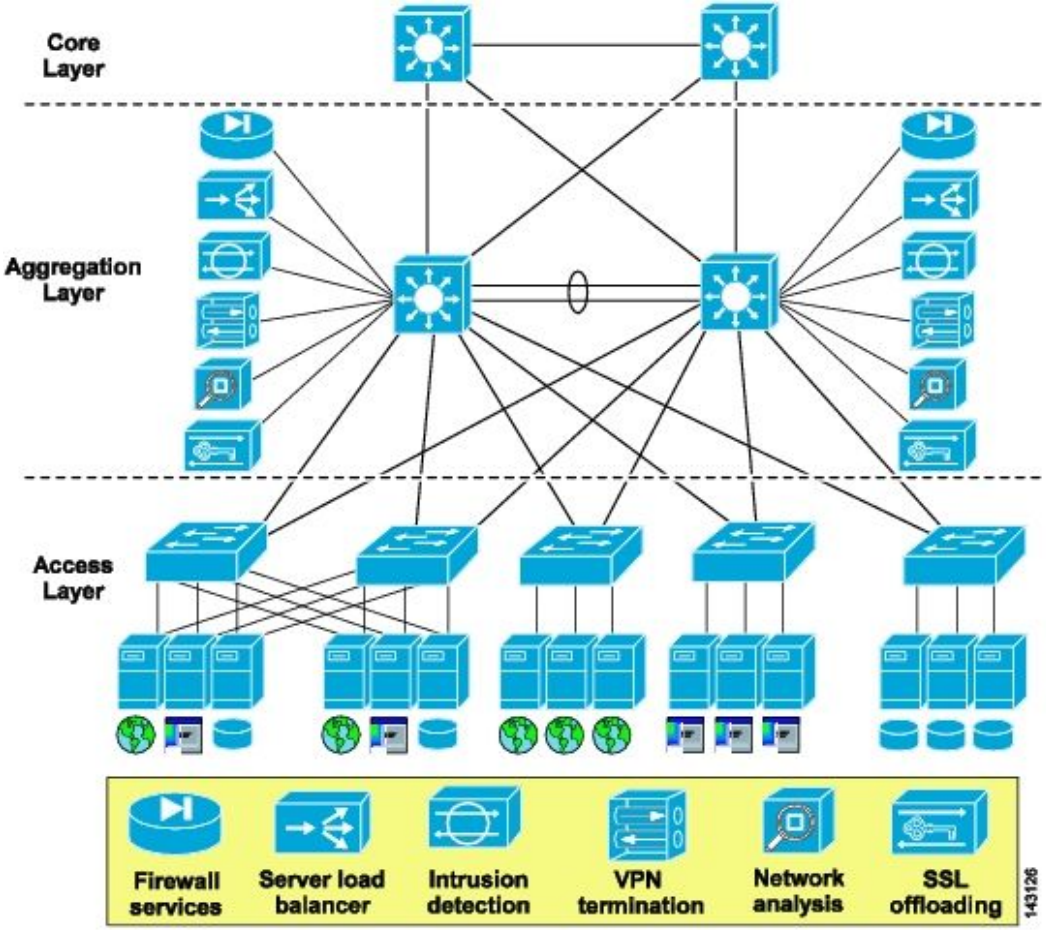
Networking 101

Switches, VLAN’s, Subnets and more..

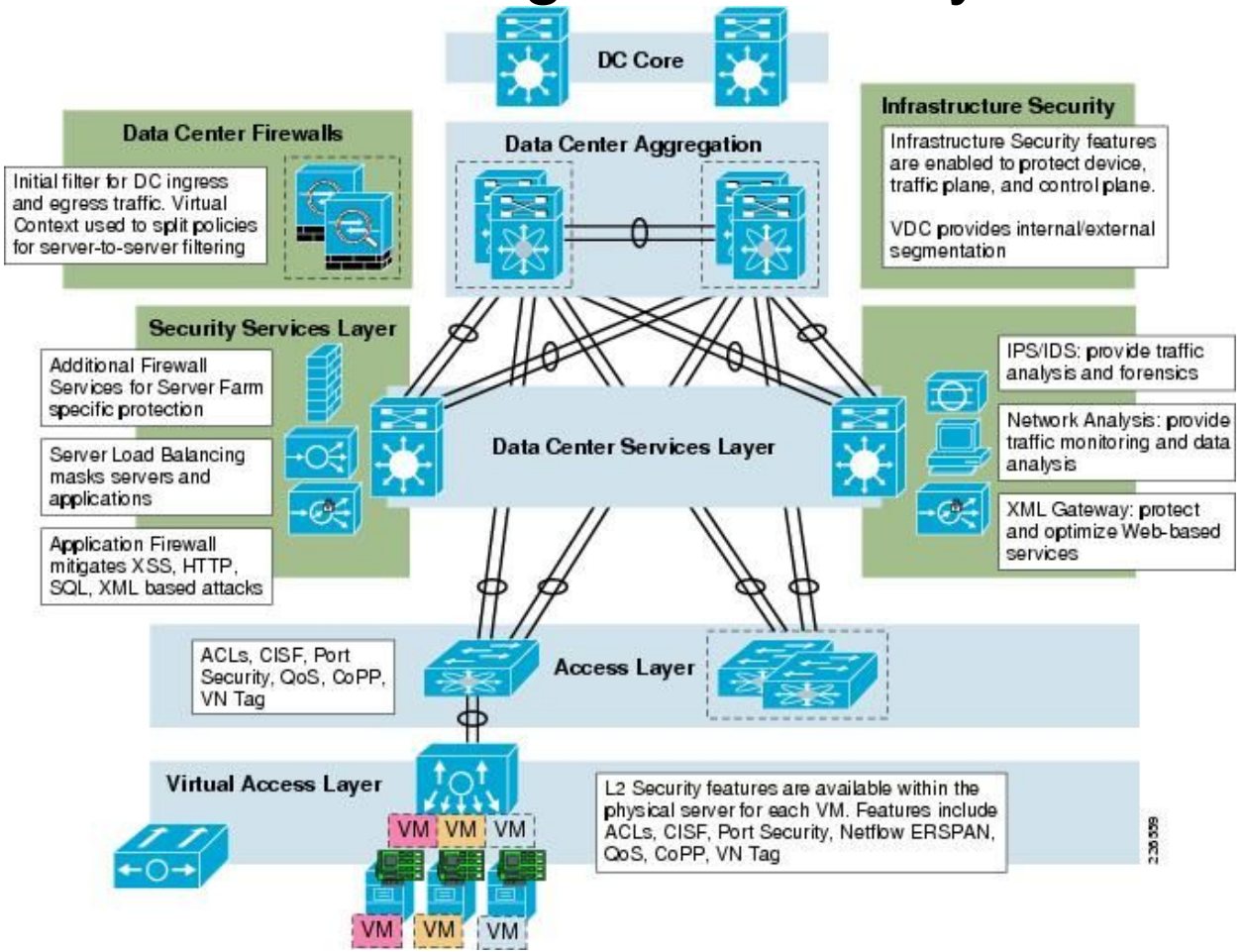
Arun Sriraman

Typical Data Center

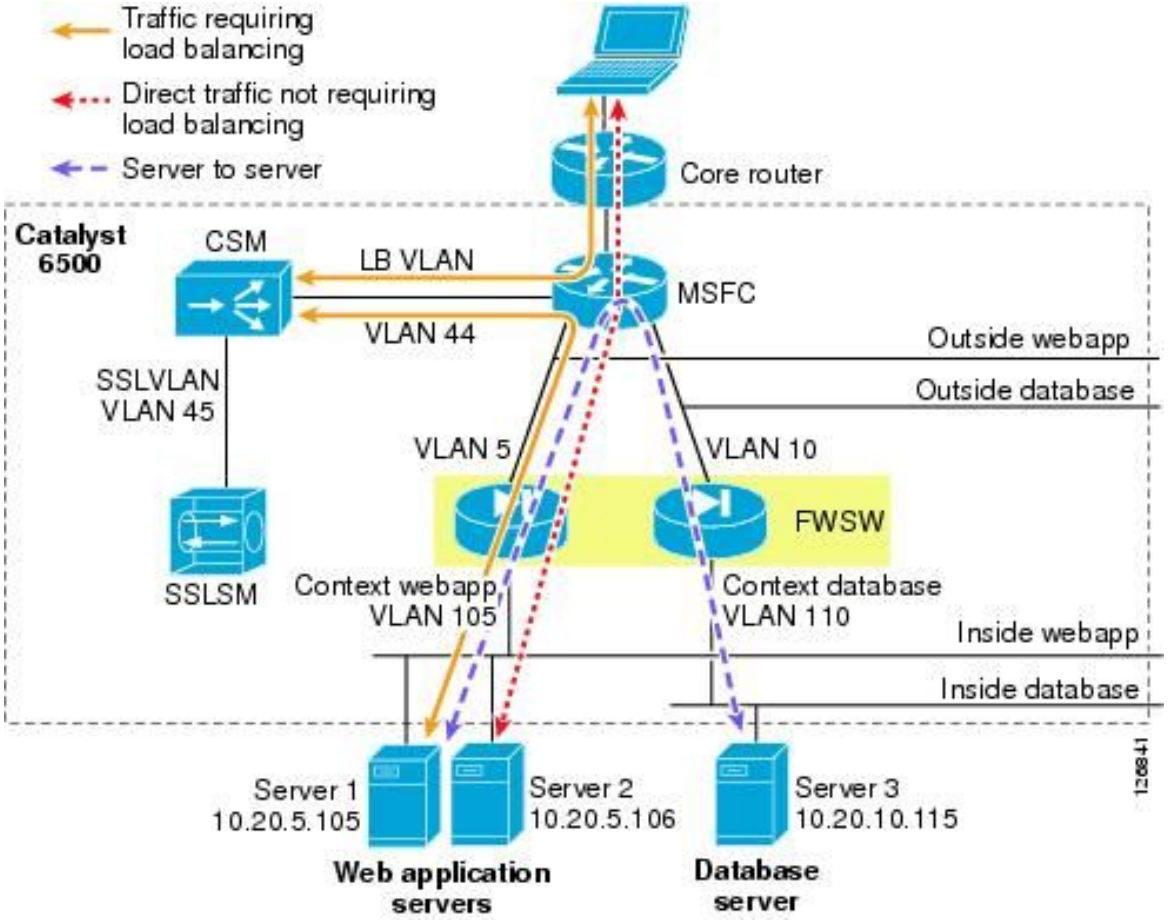


Arun Sriraman

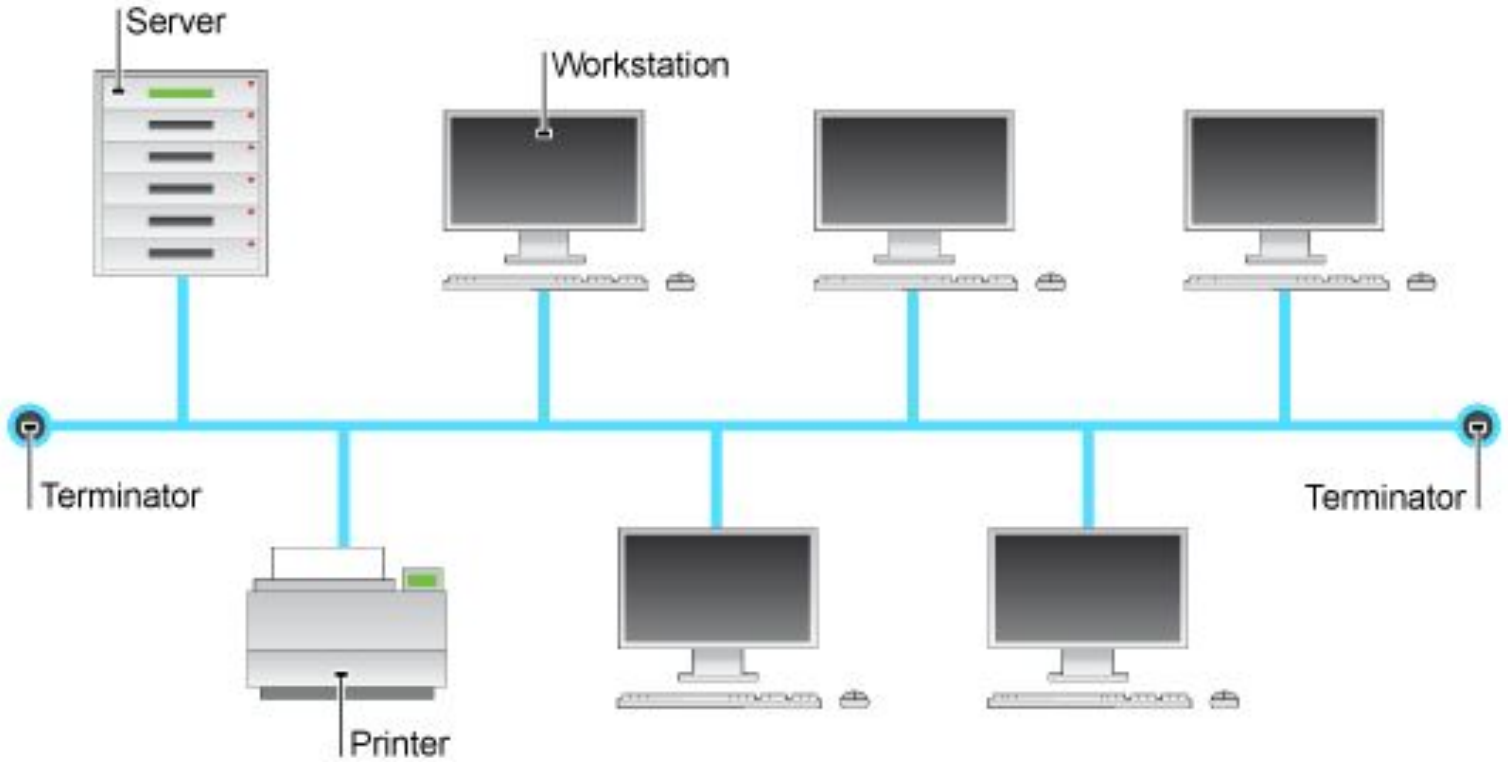
Virtualization in DC brings in more layers



Security Security Security…

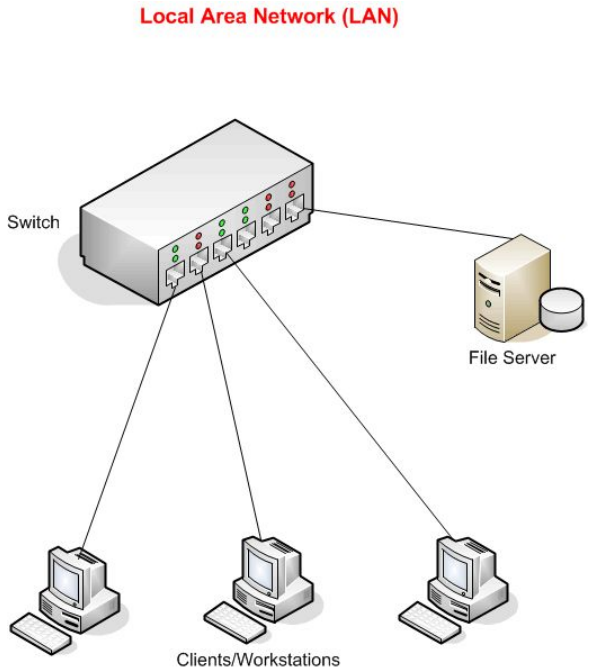


Good old days - remember token ring?



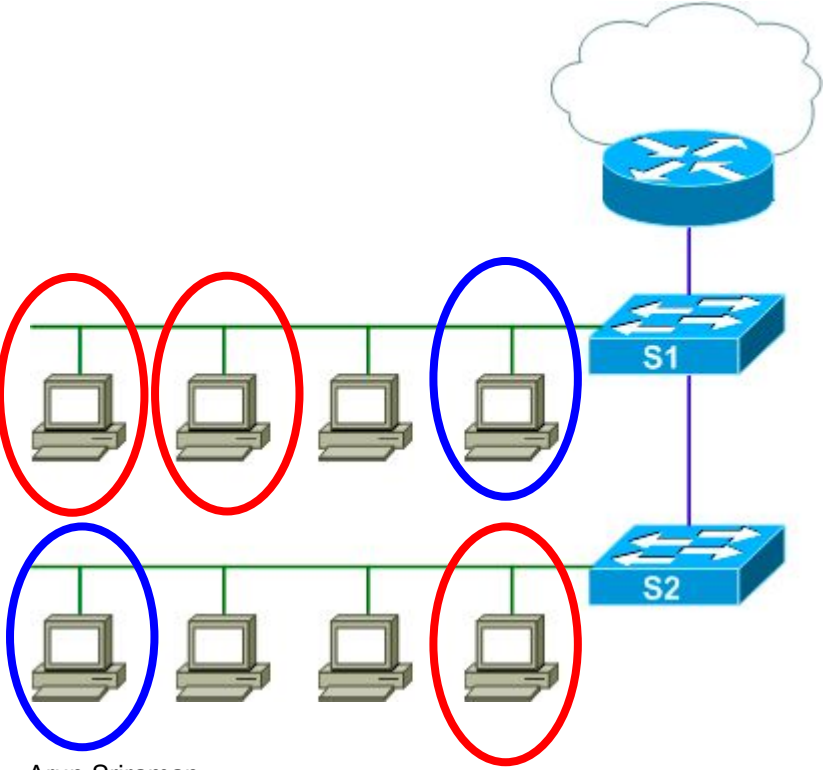
Single broadcast domain (ff:ff:ff:ff:ff:ff)

Then comes the switch

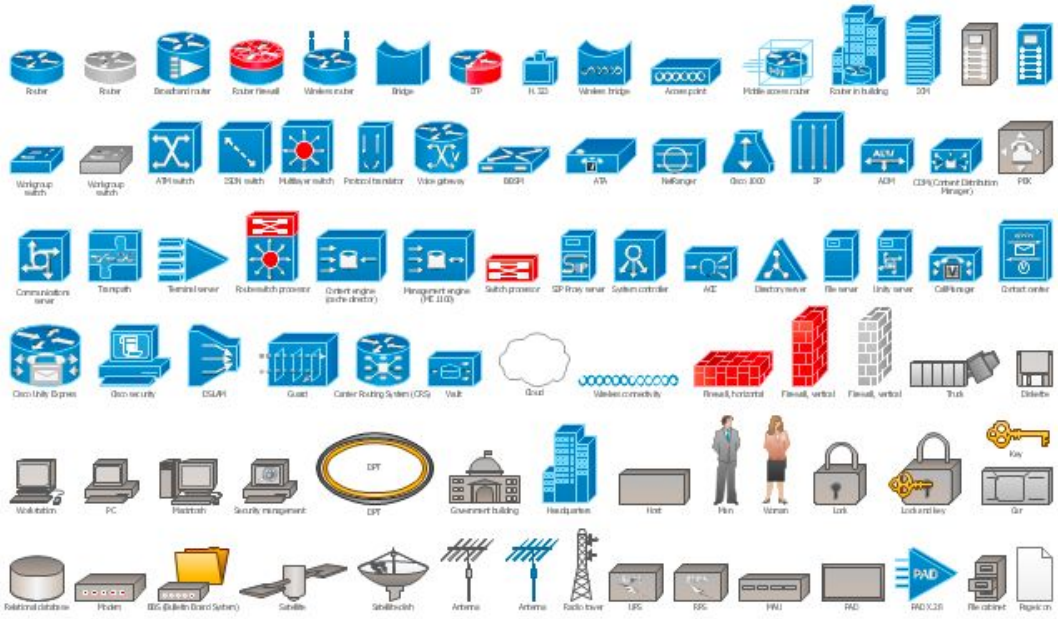


- Managed vs Unmanaged
- Collision domain - Transmit packets a little intelligently.

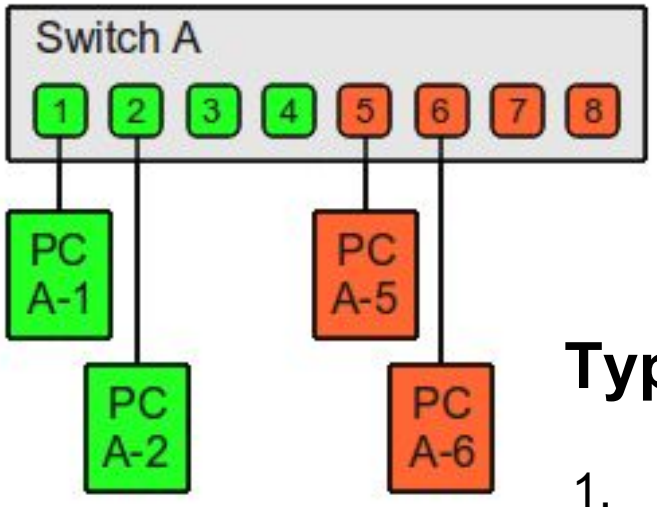
Two LAN's connected by switches.

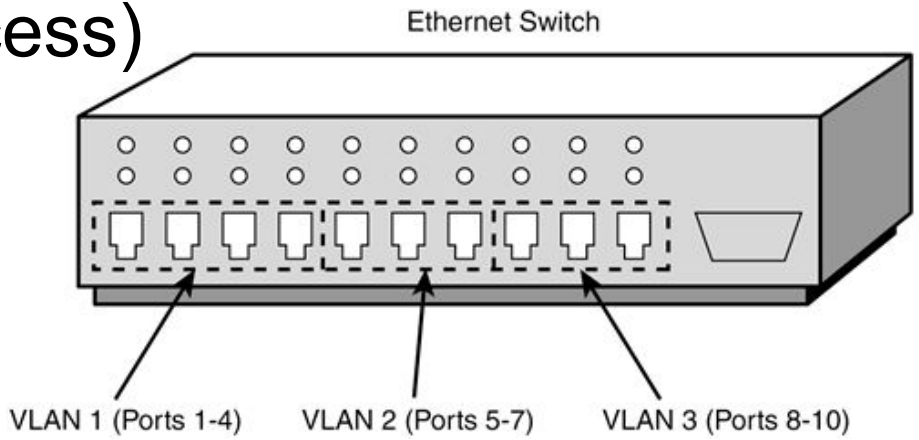


ATM Switch



VLAN's.. (port based - Access)

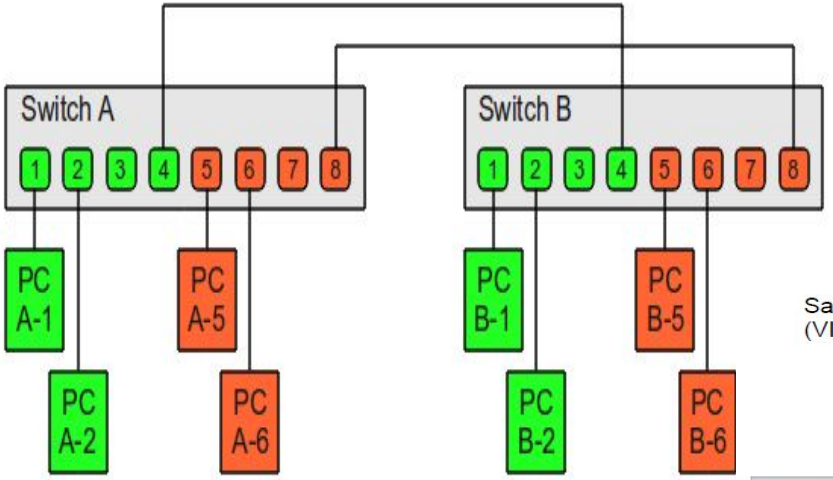




Most common deployments: 1 & 4

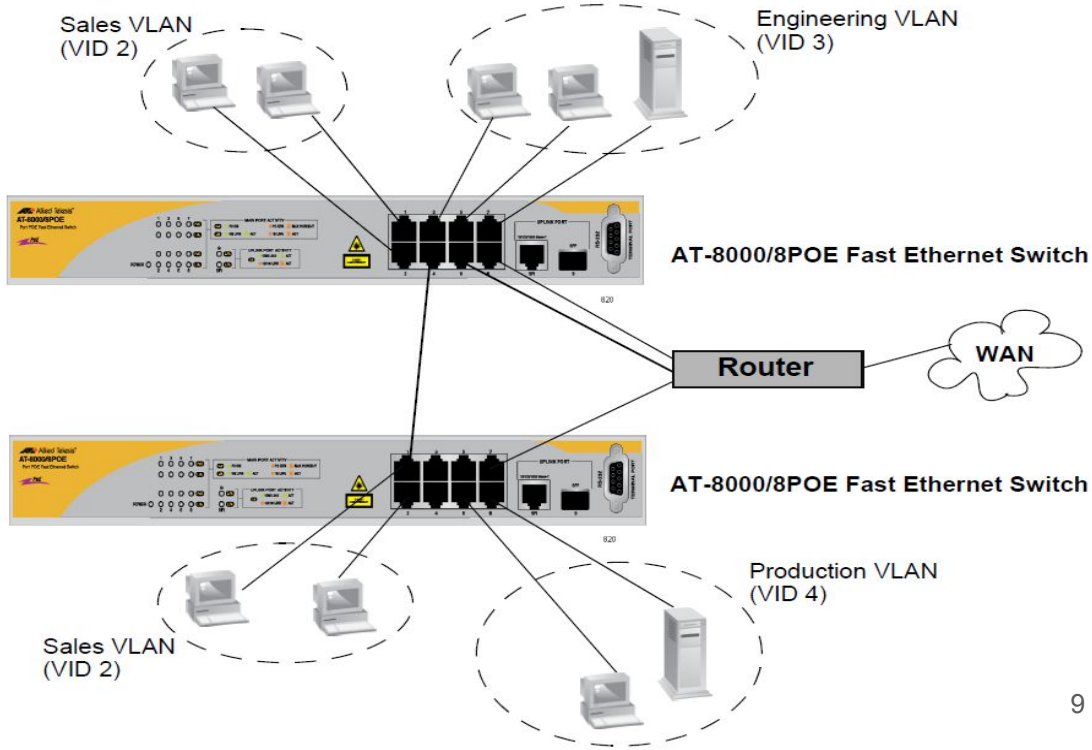
Types

1. Layer 1 VLAN: Membership by Port
2. Layer 2 VLAN: Membership by MAC Address
3. Layer 2 VLAN: Membership by Protocol Type
4. Layer 3 VLAN: Membership by IP Subnet Address
5. Higher Layer VLAN's

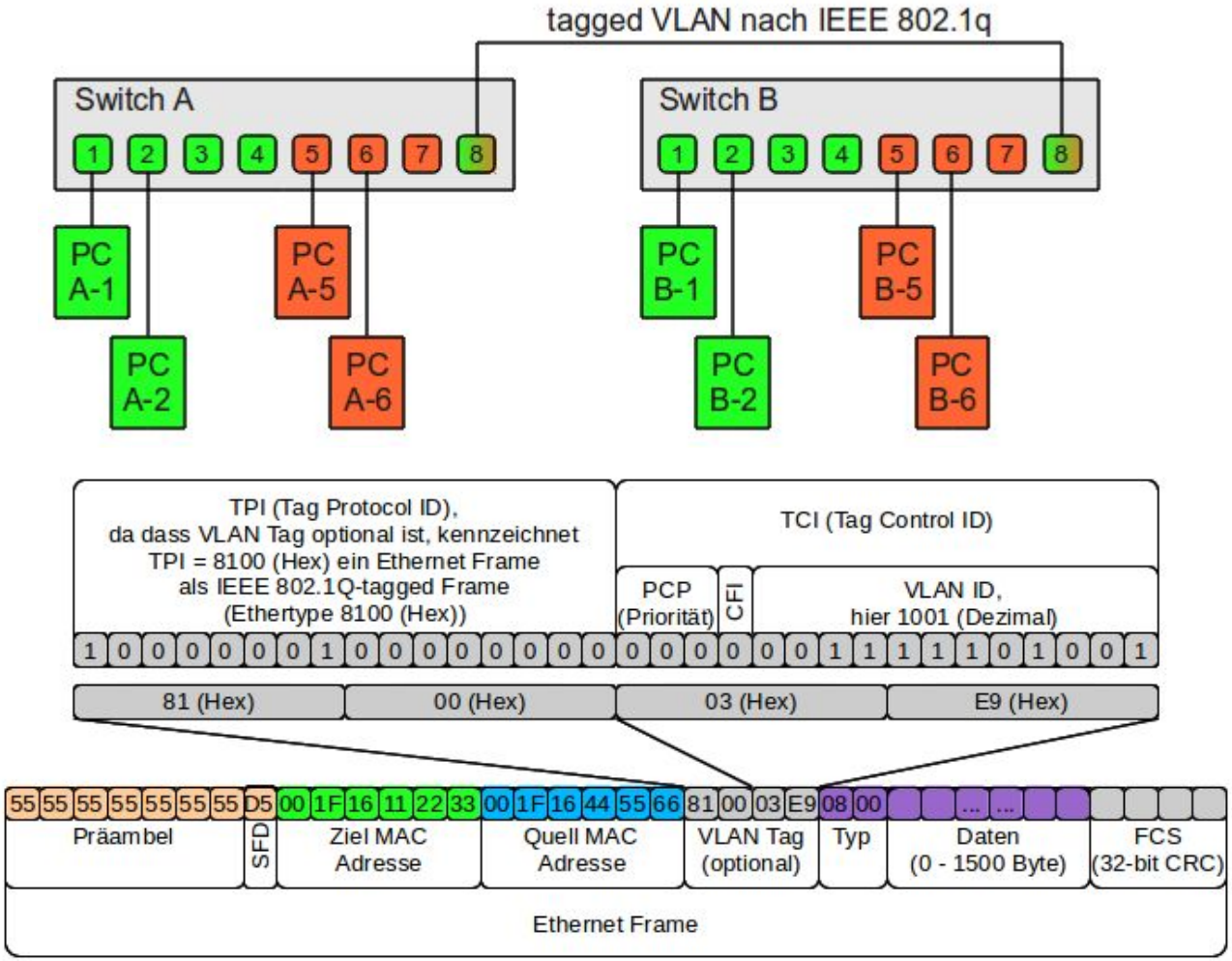


https://www.thomas-krenn.com/en/wiki/VLAN_Basics

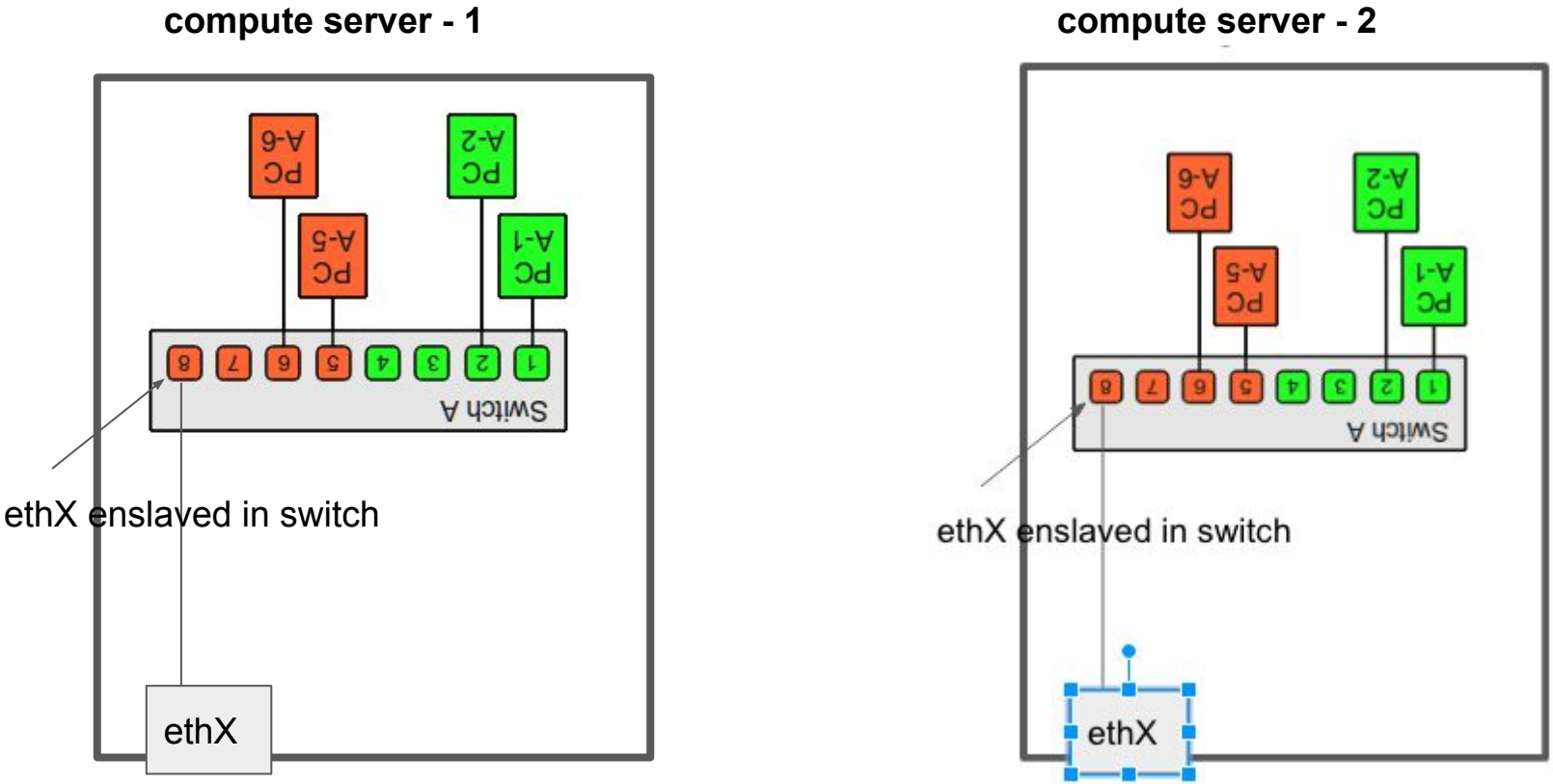
Access VLAN's



Trunking - Tagged VLAN's



Now, we go virtual...



Tagged/Untagged/Access/Native/Trunk

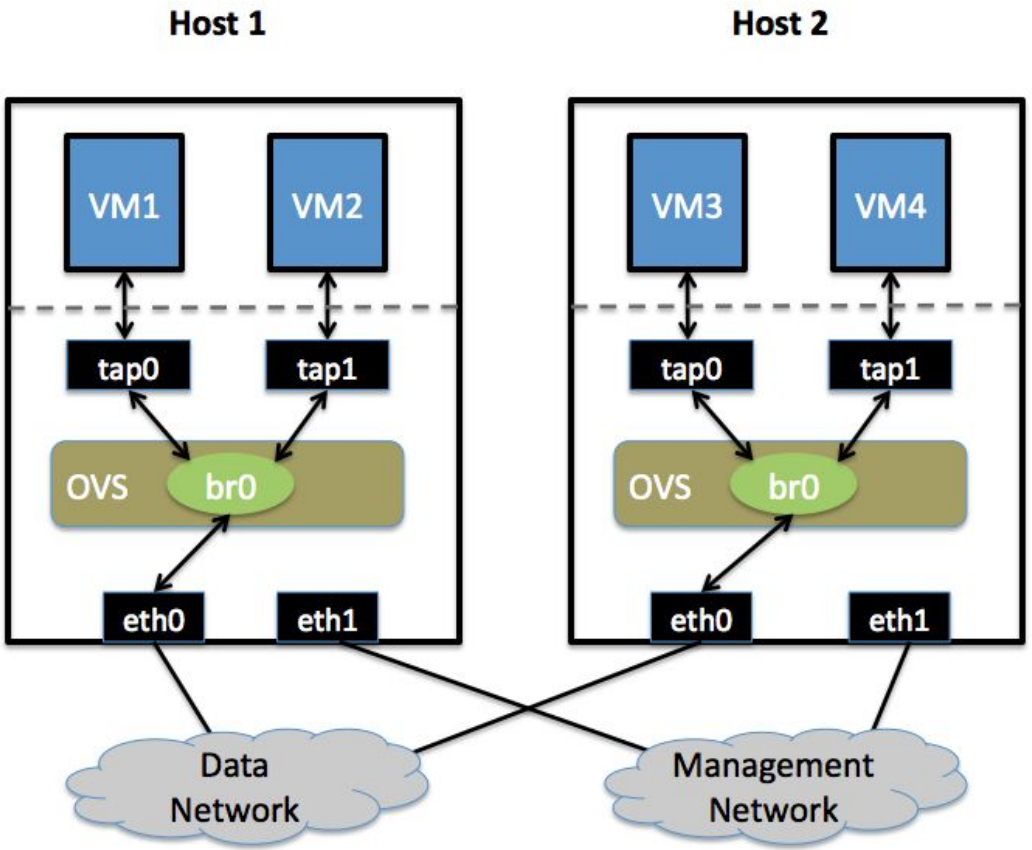
<Whiteboarding>

Example with OVS

Where should the tags go?

Use ovs-vsctl to add tags to the tap interfaces that sit inside OVS.

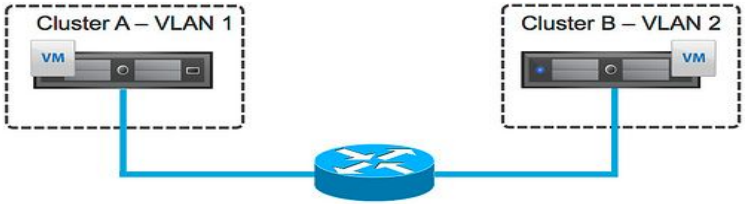
```
ovs-vsctl add-br br0
ovs-vsctl add-port br0 eth0
ovs-vsctl add-port br0 tap0 tag=100
```



http://openvswitch.org/support/config-cookbooks/vlan-configuration-cookbook/

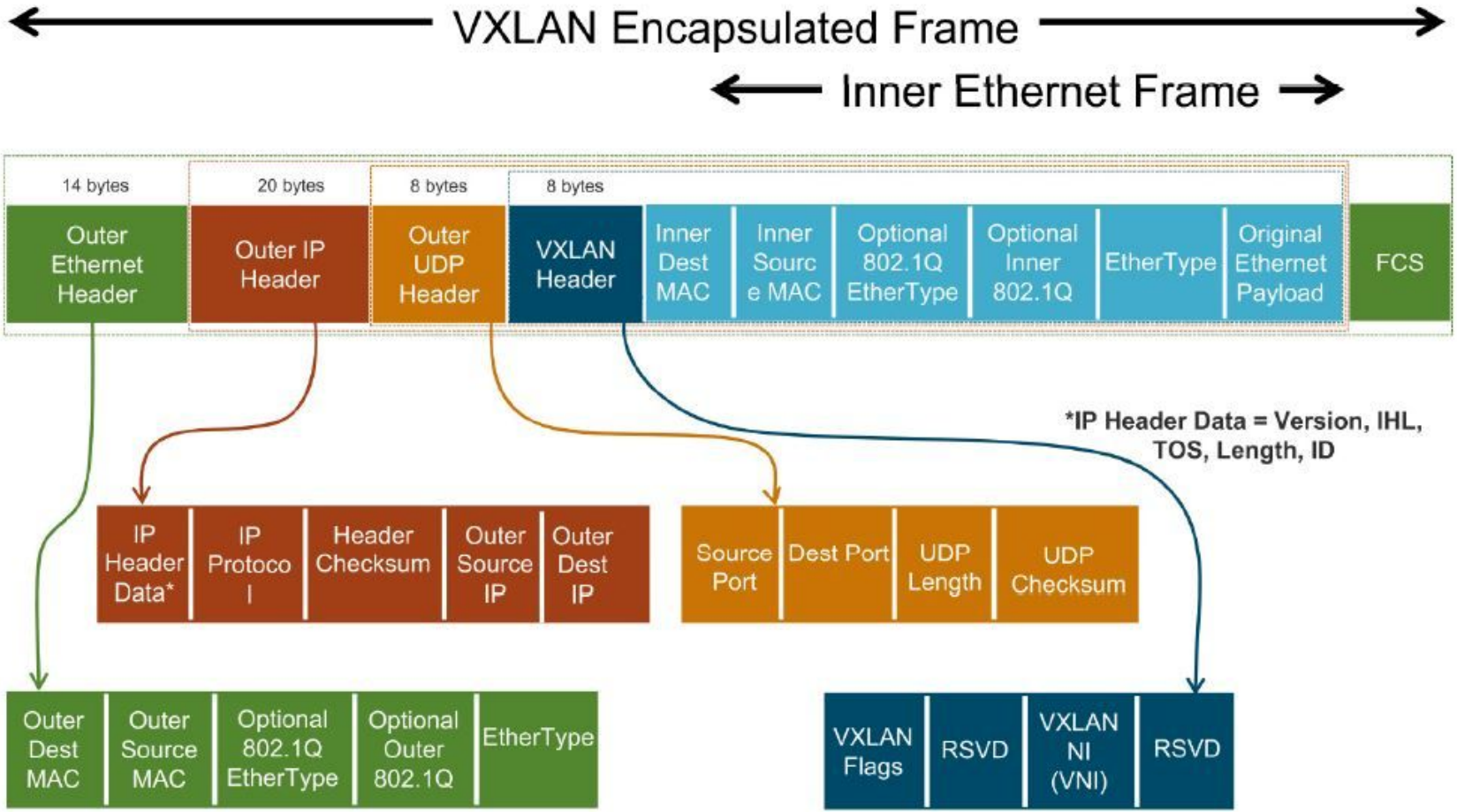
Other networking types? (Why VXLAN)

Communicate from one VLAN network to another?

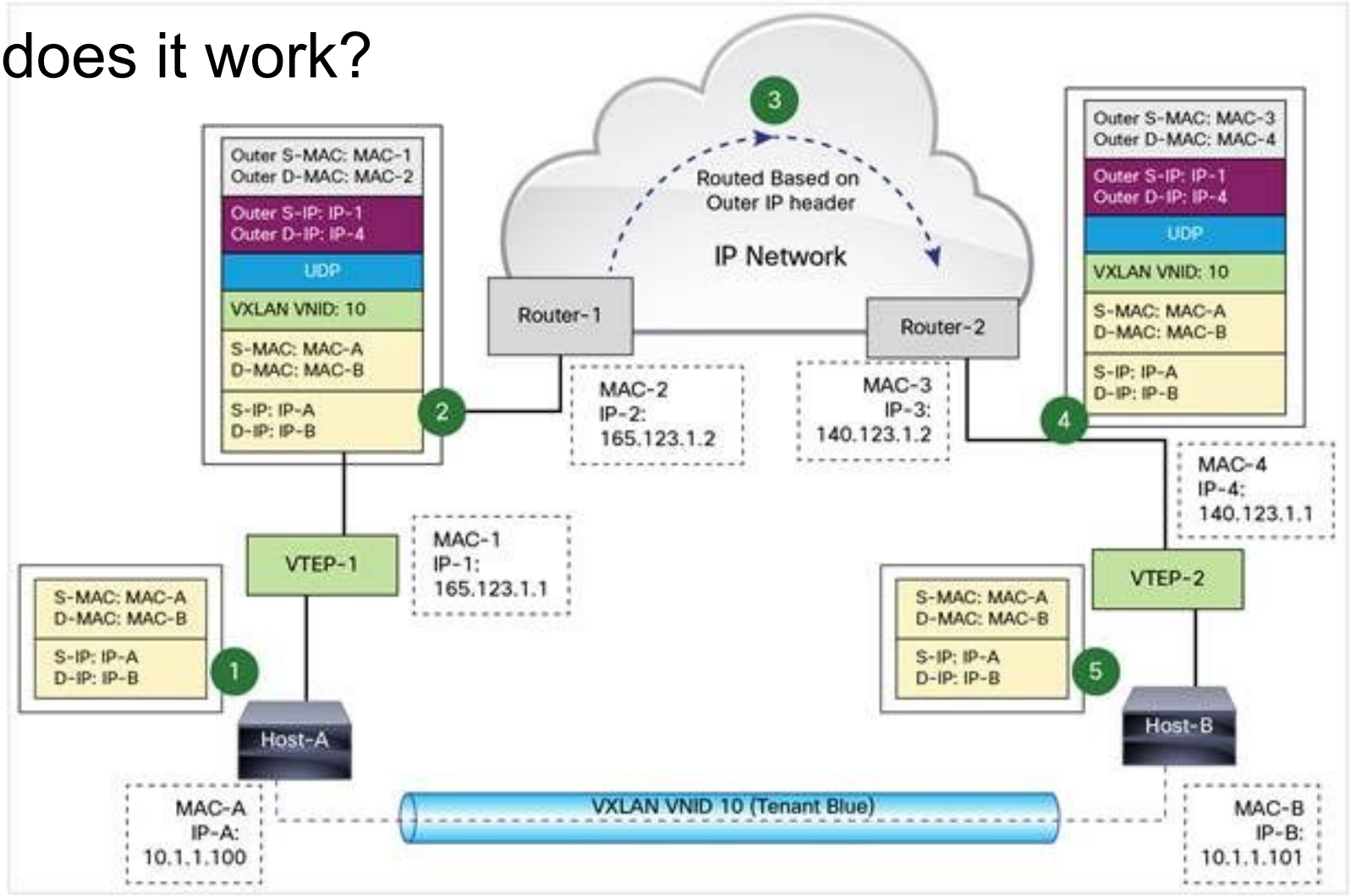


Lets go up (Layer 3 - Use a router)...
But what if you want the two VM's to be on the same domain? (similar to a VLAN)

Lets use VXLAN!

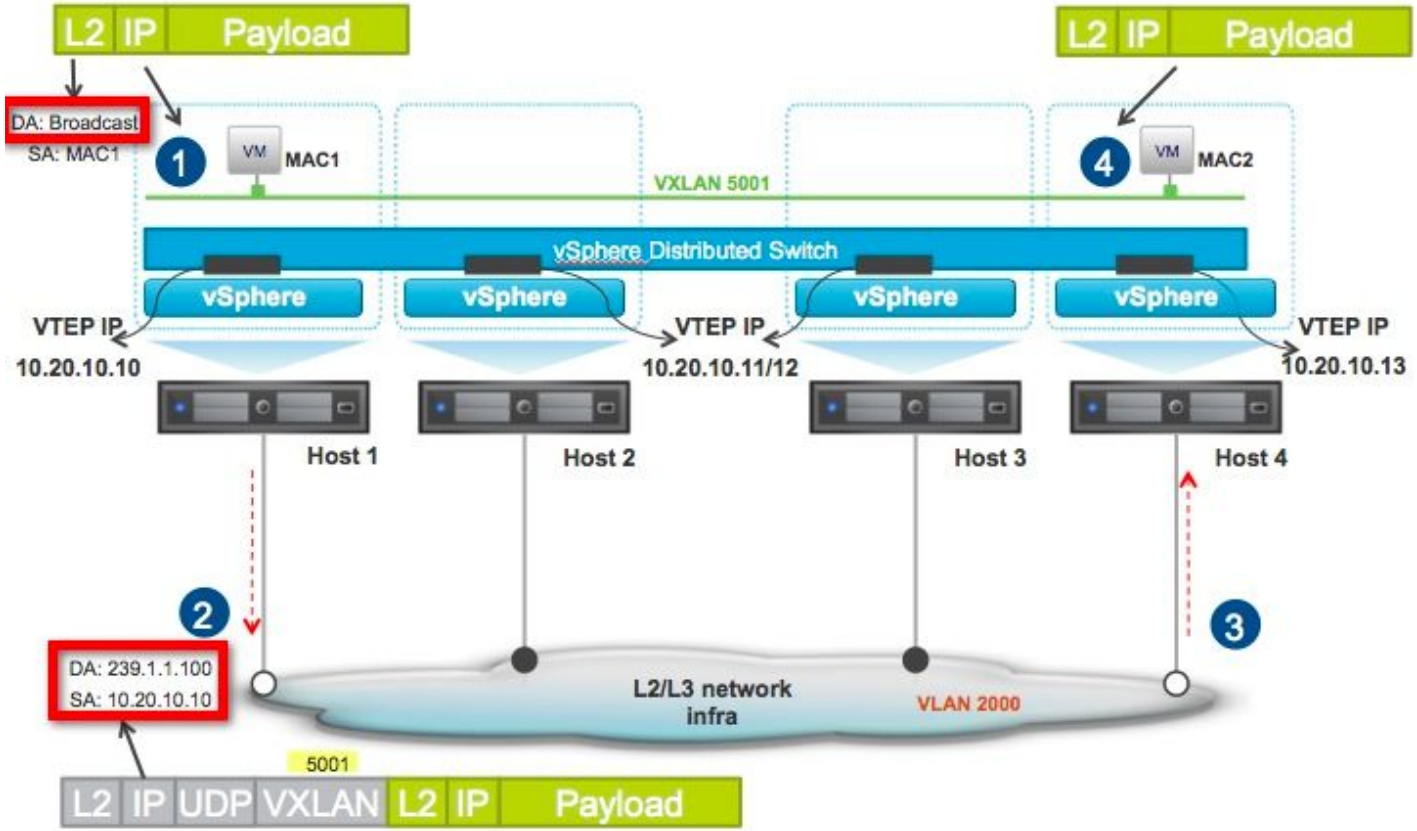

VXLAN Encapsulated Frame
Inner Ethernet Frame
14 bytes
20 bytes
8 bytes
8 bytes
Outer Ethernet Header
Outer IP Header
Outer UDP Header
VXLAN Header
Inner Dest MAC
Inner Source MAC
Optional 802.1Q EtherType
Optional Inner 802.1Q
EtherType
Original Ethernet Payload
FCS
*IP Header Data = Version, IHL, TOS, Length, ID
IP Header Data*
IP Protocol
Header Checksum
Outer Source IP
Outer Dest IP
Source Port
Dest Port
UDP Length
UDP Checksum
Outer Dest MAC
Outer Source MAC
Optional 802.1Q EtherType
Optional Outer 802.1Q
EtherType
VXLAN Flags
RSVD
VXLAN NI (VNI)
RSVD

How does it work?



VXLAN Use Cases

Natively supports multicasting

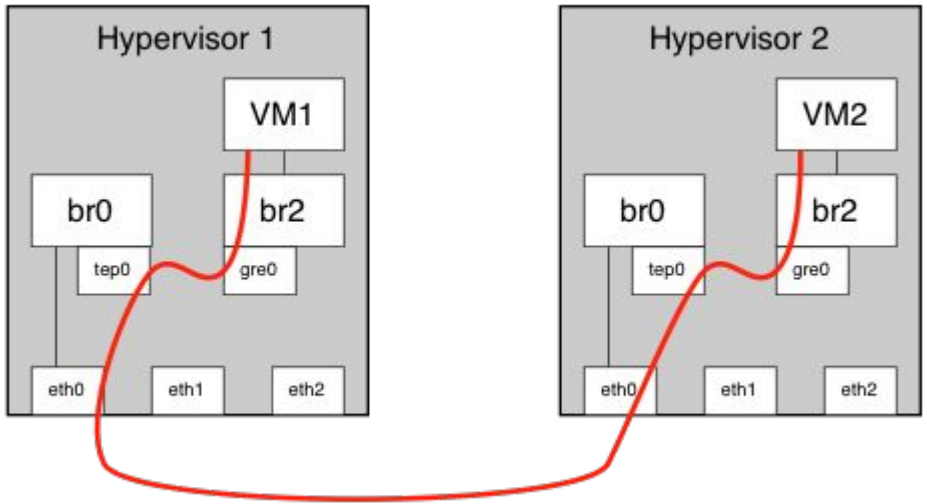


What's GRE and why?

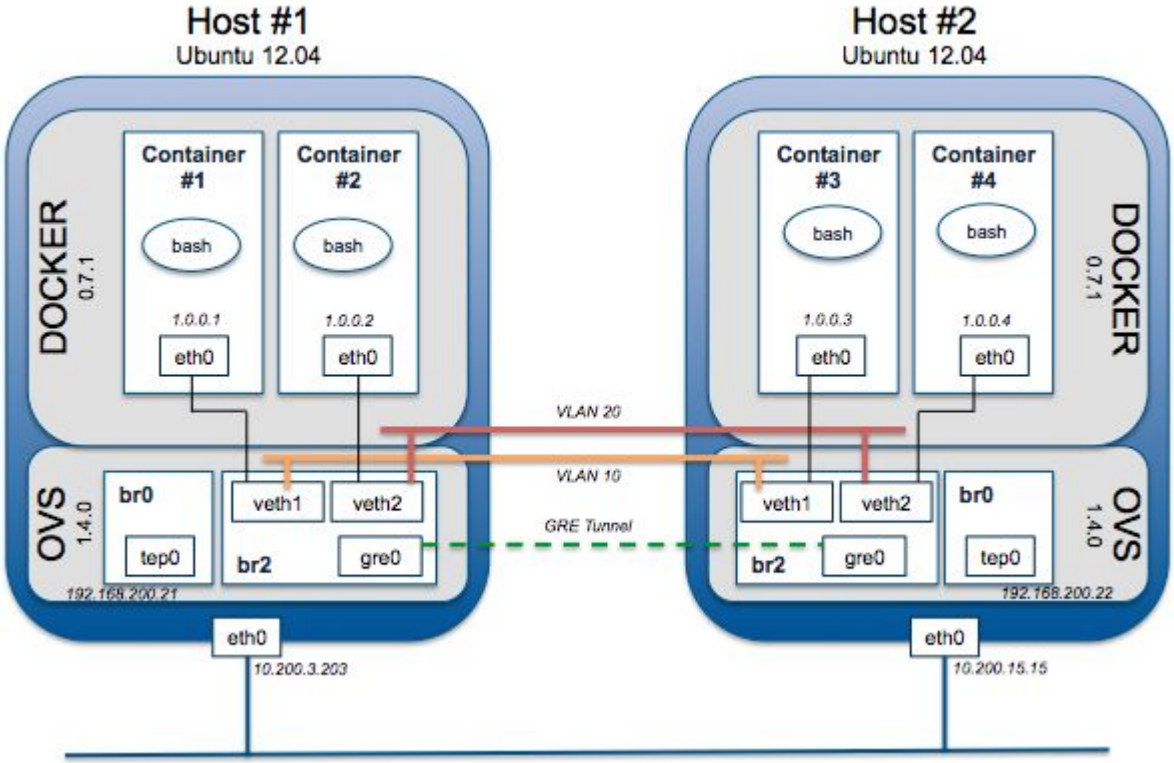
Both VXLAN & GRE are encap methods.

VXLAN uses UDP -> 5 tuple routing (UDP source and dest ports)

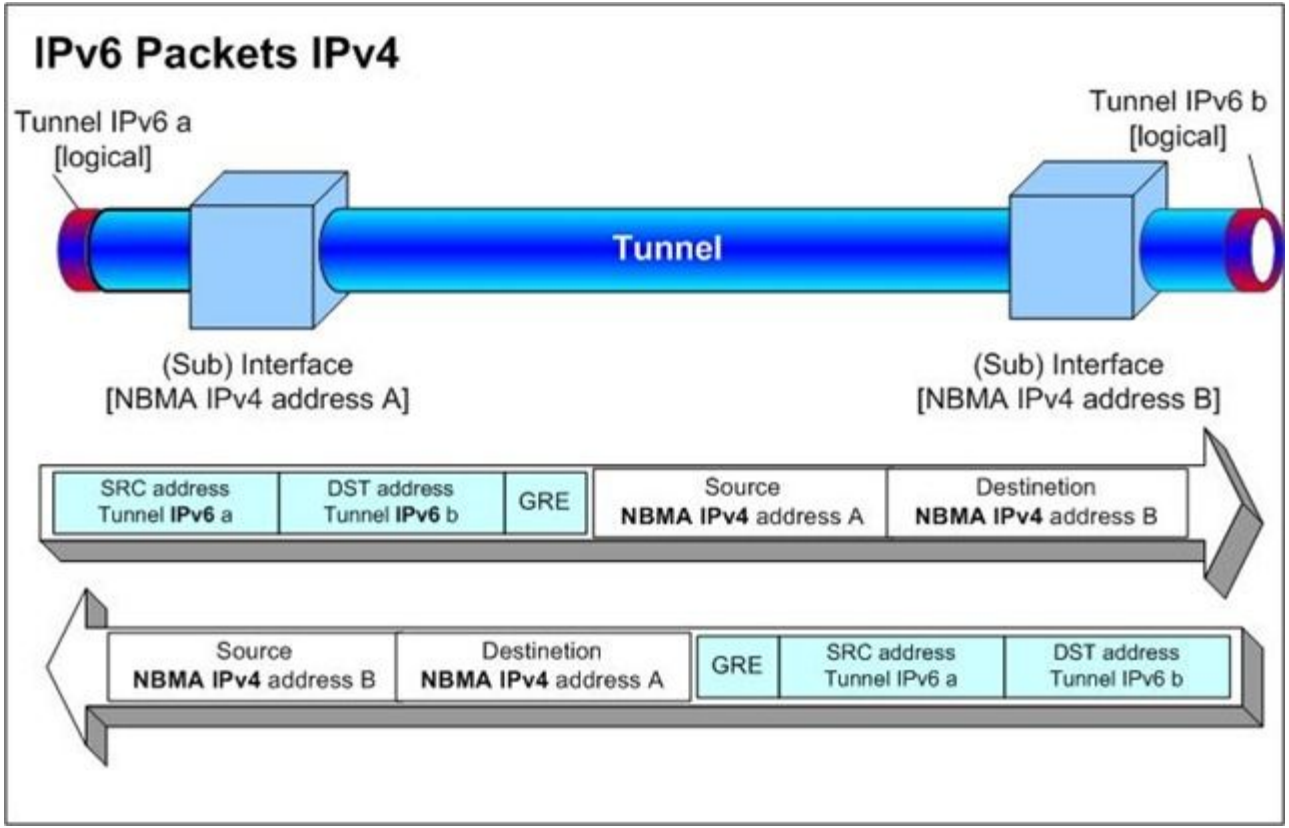
GRE -> needs specific hardware to achieve routing performance.



Docker/Openstack setup?



GRE Use Case - Protocol encap?

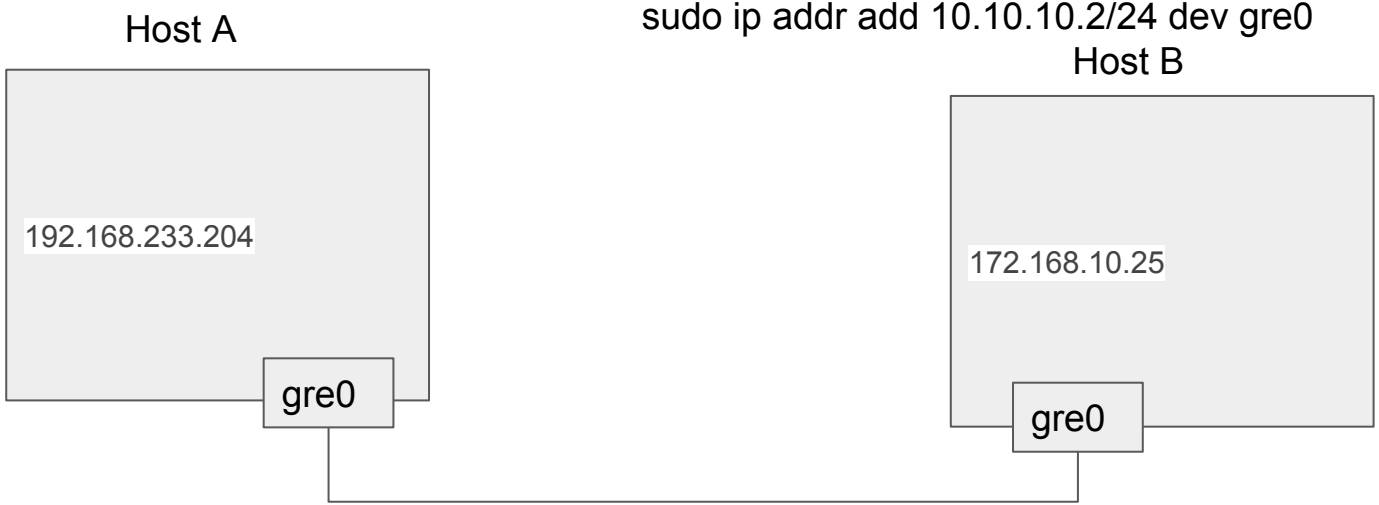


VXLAN can do it too!! (Think VXLAN + MPLS)

Setting up a GRE Tunnel on linux

linux module : ip_gre

sudo ip tunnel add gre0 mode gre remote 192.168.233.204 local 172.168.10.25 ttl 255
sudo ip link set gre0 up
sudo ip addr add 10.10.10.2/24 dev gre0



sudo ip tunnel add gre0 mode gre remote 172.168.10.25 local 192.168.233.204 ttl 255
sudo ip link set gre0 up
sudo ip addr add 10.10.10.1/24 dev gre0

Other protocols

SSH Tunnel

ssh -f <user>@<host> -L <local_port>:<host>:<remote_port>
Eg: for our rabbit database:
ssh -i ~/Downloads/id_rsa -L :15672:localhost:15672 root@pf9.platform9.net

VPN

HTTP

ICMP Tunnels

Troubleshooting

route -n

wireshark & tcpdump (tcpdump -i <interface> -vv vlan <vlan_id>)

ping & traceroute

ovs-vsctl show

ovs-ofctl dump-flows

brctl show / brctl showmacs

/sys/class/net/*

/proc/net/*

iptables

ip netns

arp -n

ebtables

Linux bonds

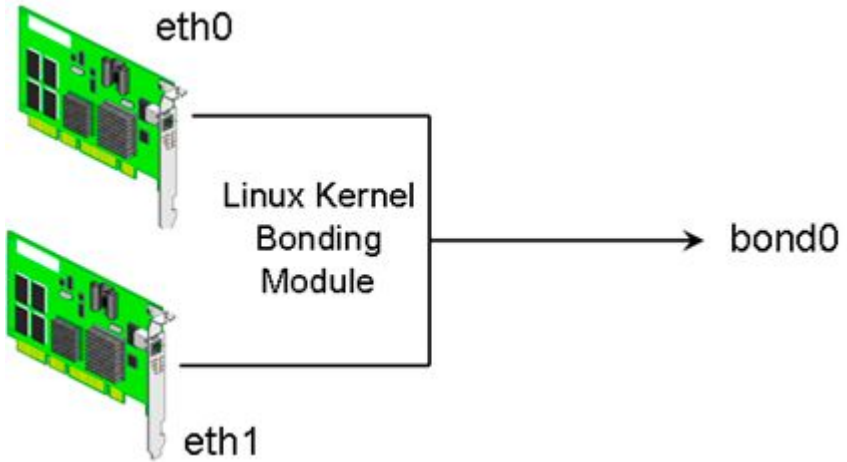
6 modes. Most common:

HA - Active Passive

LACP - Port aggregation

- ALB (Adaptive load balancing)

```
ifconfig bond0 <ip> up
ifenslave bond0 eth0
ifenslave bond0 eth1
```



Linux kernel documentation is a great place to look -
https://www.kernel.org/pub/linux/kernel/people/marcelo/linux-2.4/Documentation/networking/bonding.txt