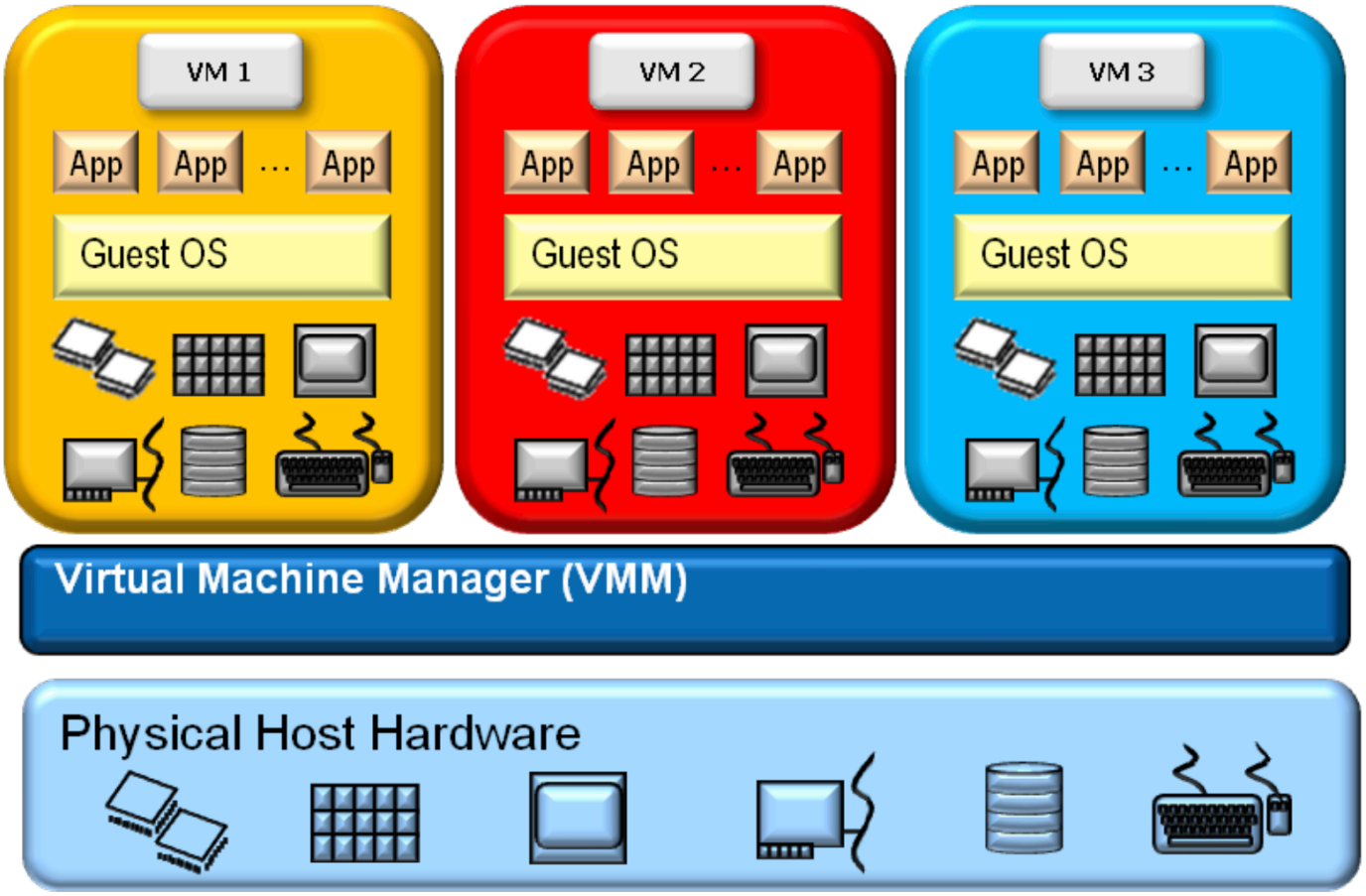
Network Virtualization and Host SDN

ECE/CS598HPN

Multi-tenant datacenters



Networking between VMs

- Conventional approach:
  - Physical network treats each VM as a host directly attached to it.
  - The vSwitch in the hypervisor extends physical network.

Issues with this approach

- Physical network is aware of tenant addresses.
  - Difficult to scale.

- Address space tied to physical network.
  - VMs get IP from subnet of first L3 router.
  - Hinders VM mobility.
  - VMs can't run their on IP address management schemes.

Networking between VMs

- Different workloads (VM clusters) have different service requirements.
  - Some require L2 routing.
  - Some require L3 routing.
  - Some require special services (e.g. L4 load balancing).

Virtualization techniques

- VLANs: virtualized L2 domains.
- VRF: virtualized L3 forwarding tables.
- NAT: virtualized IP address space.
- MPLS: virtualized paths.

Point solutions that virtualize singular aspects.

Need for a more holistic and global approach.

Network Virtualization

Allows creating virtual networks

(each with independent service models, topologies, and addressing schemes)

over the same physical network.

These virtual networks are created, configured, and managed via global abstractions.

Network Virtualization

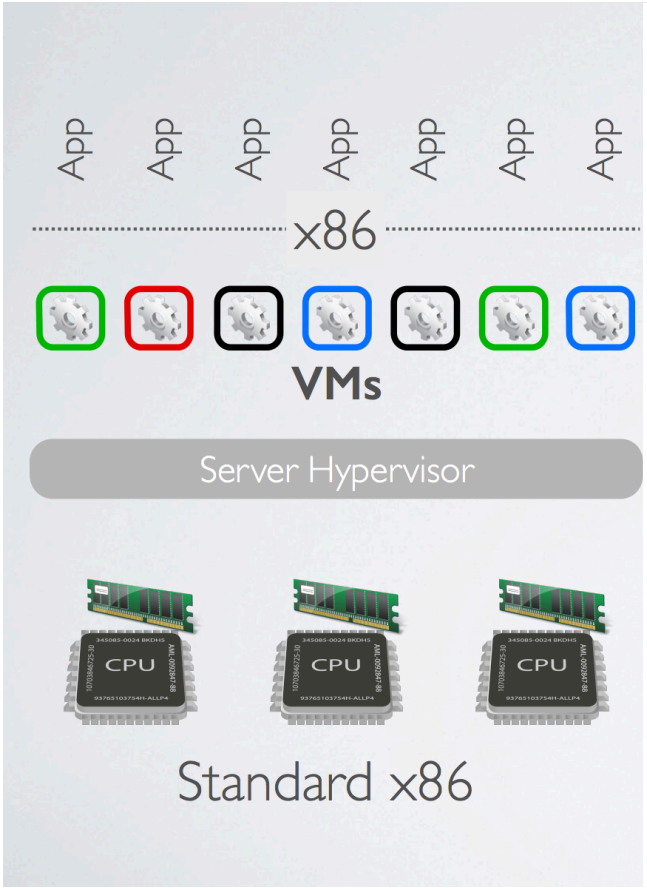
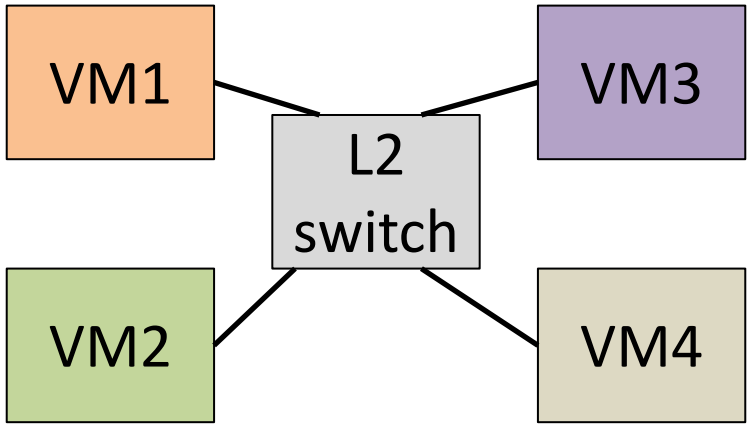
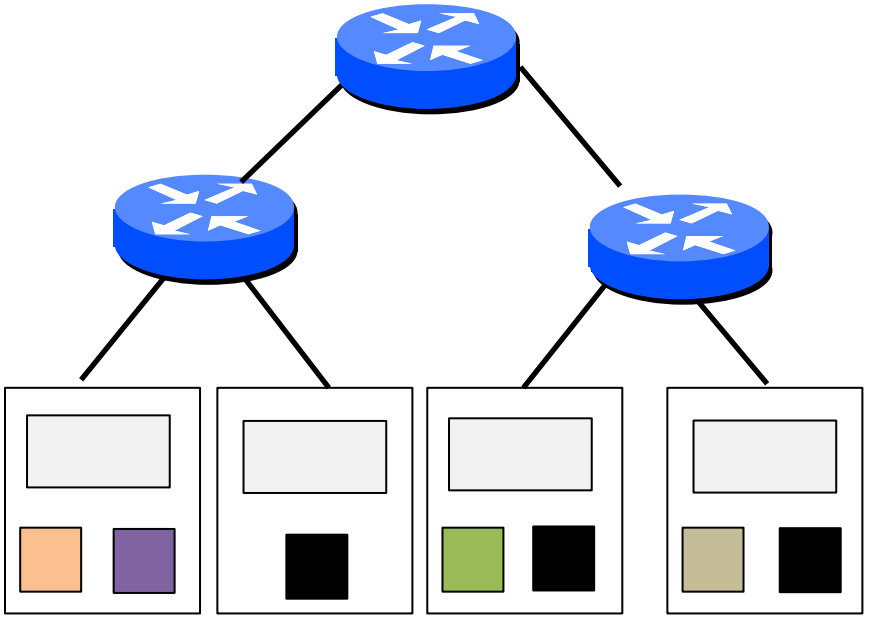


Figure from NSDI'14 talk on "Network Virtualization in Multi-tenant Datacenters"

Network Virtualization

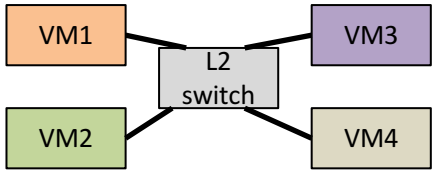


Abstraction

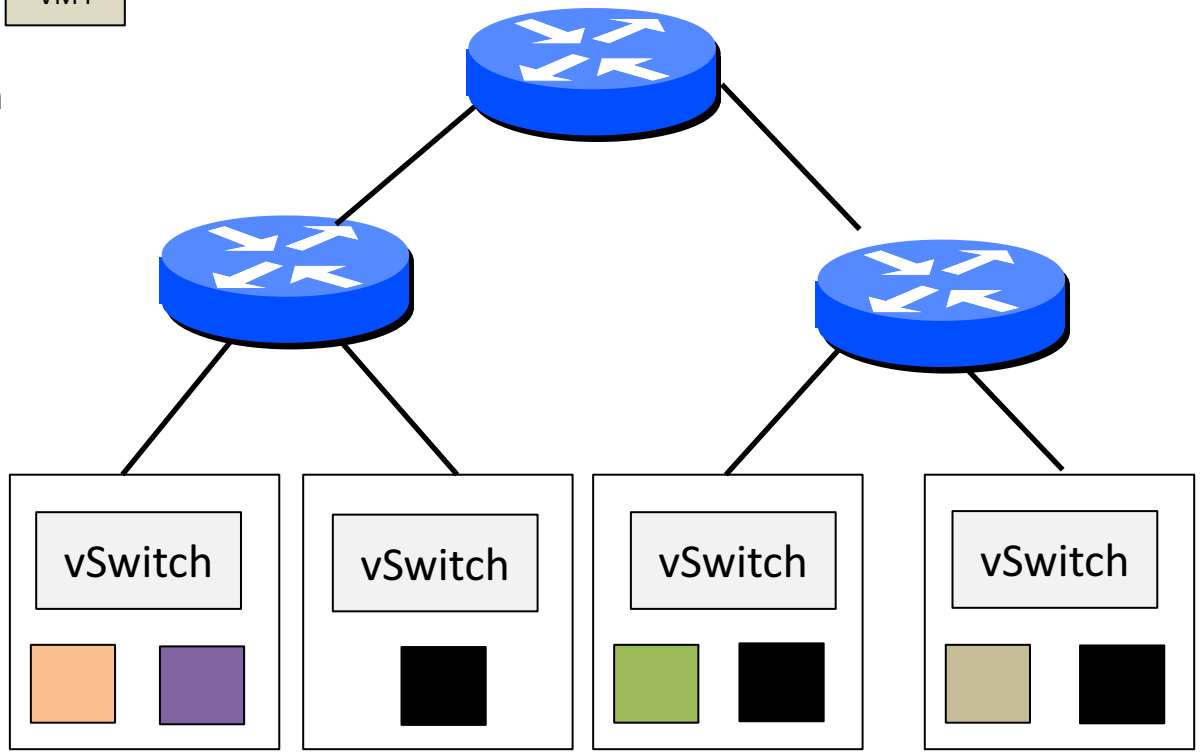


Physical Topology

vSwitches provide the abstraction



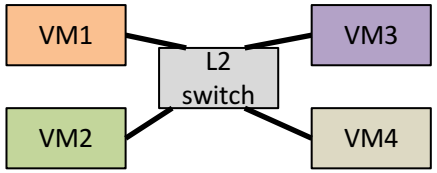
Abstraction



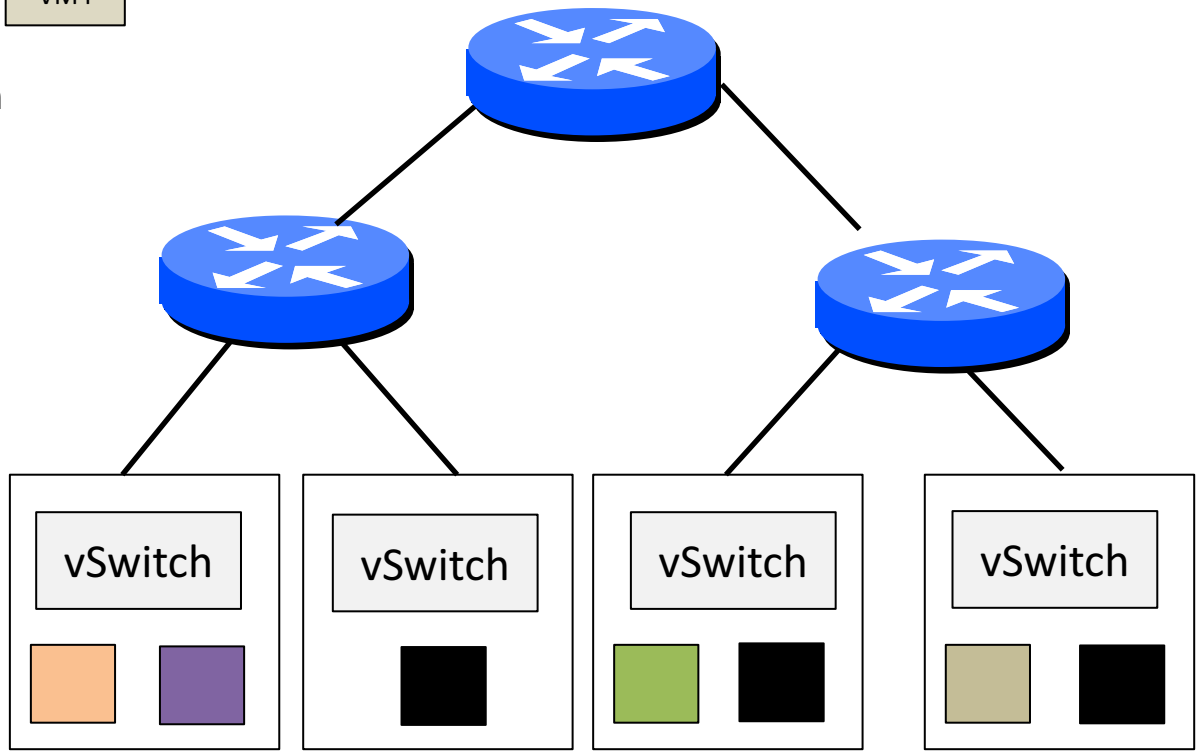
Physical Topology

vSwitches provide the abstraction

Tenants can configure logical datapaths on vSwitches using OpenFlow.



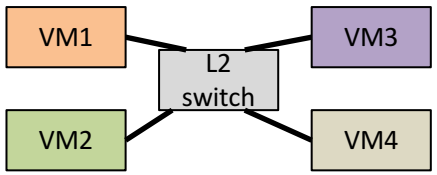
Abstraction



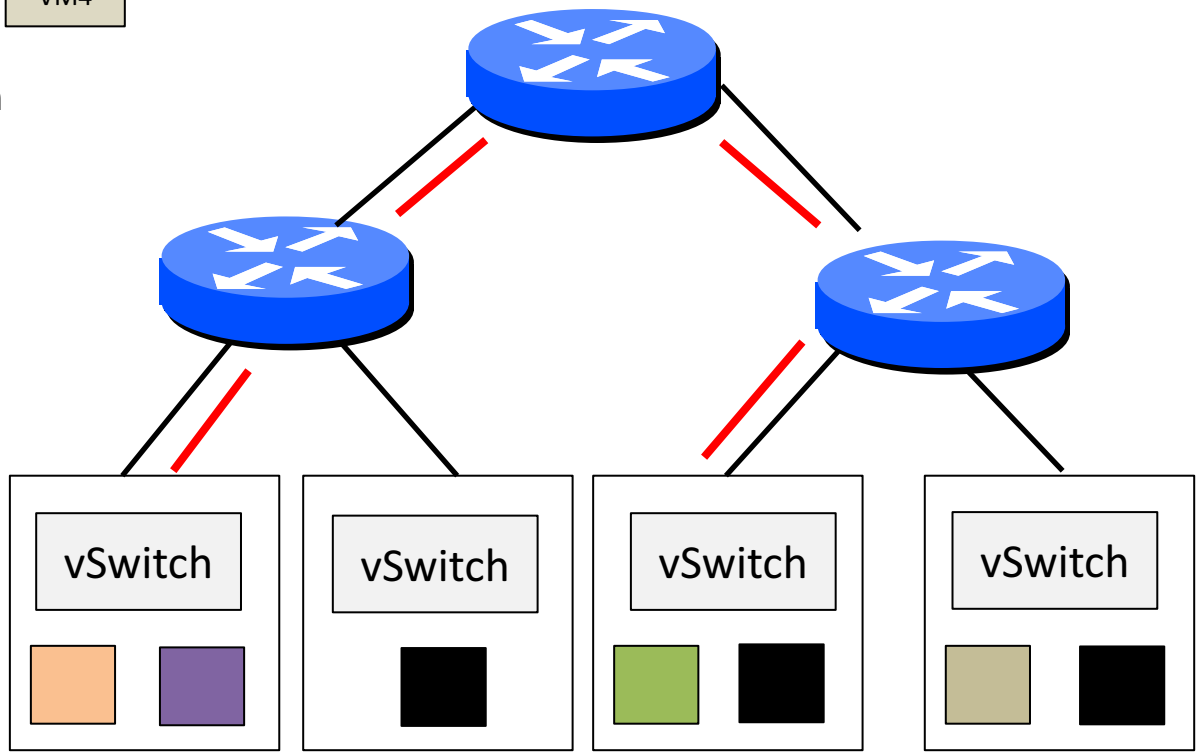
Physical Topology

vSwitches provide the abstraction

vSwitches transparently tunnel packets between VMs on different servers.

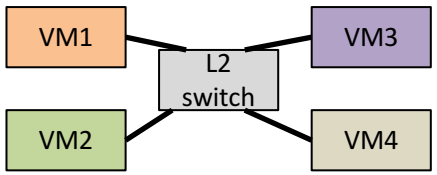


Abstraction



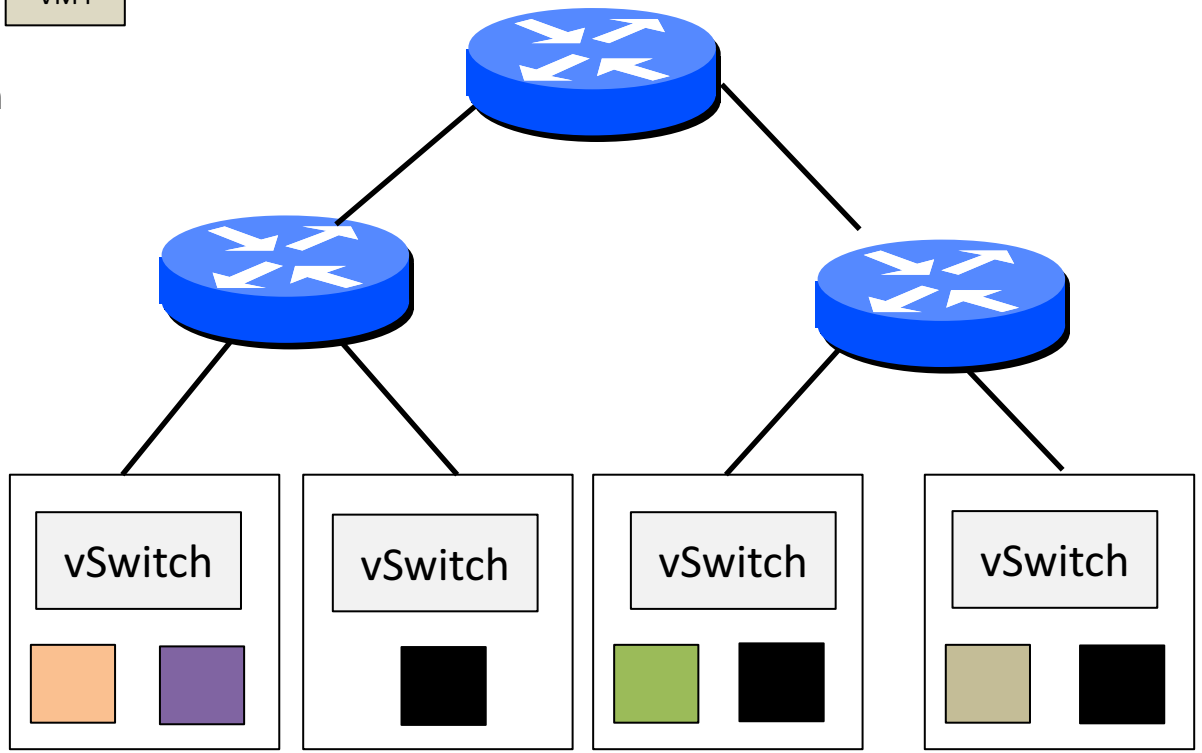
Physical Topology

vSwitches provide the abstraction



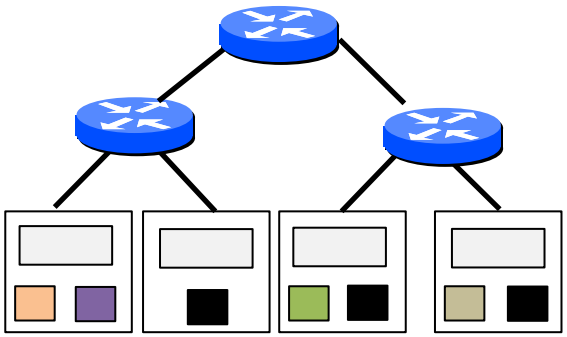
Abstraction

Physical switches simply route packets from one server to another using IP-routing.

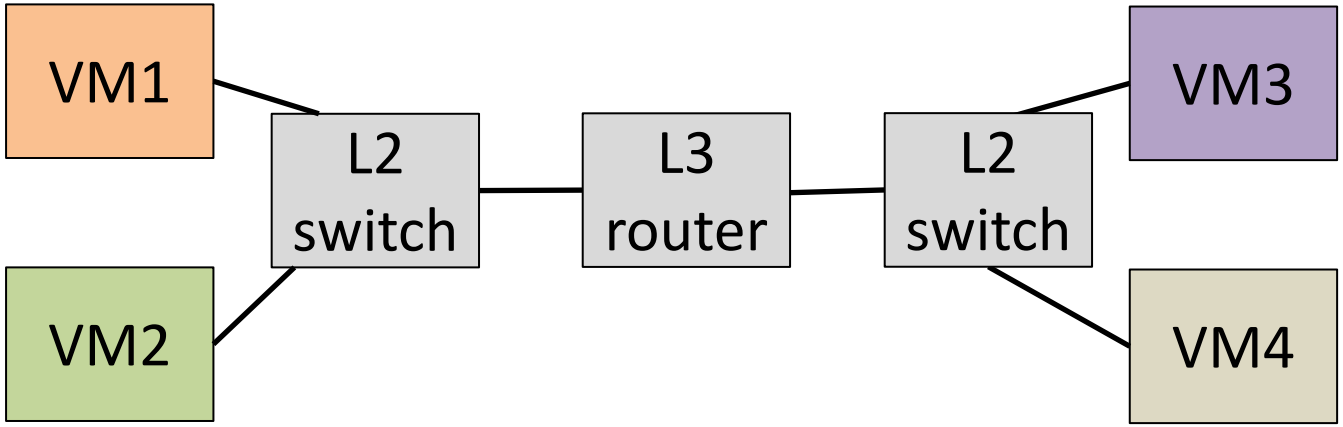


Physical Topology

More complex datapaths

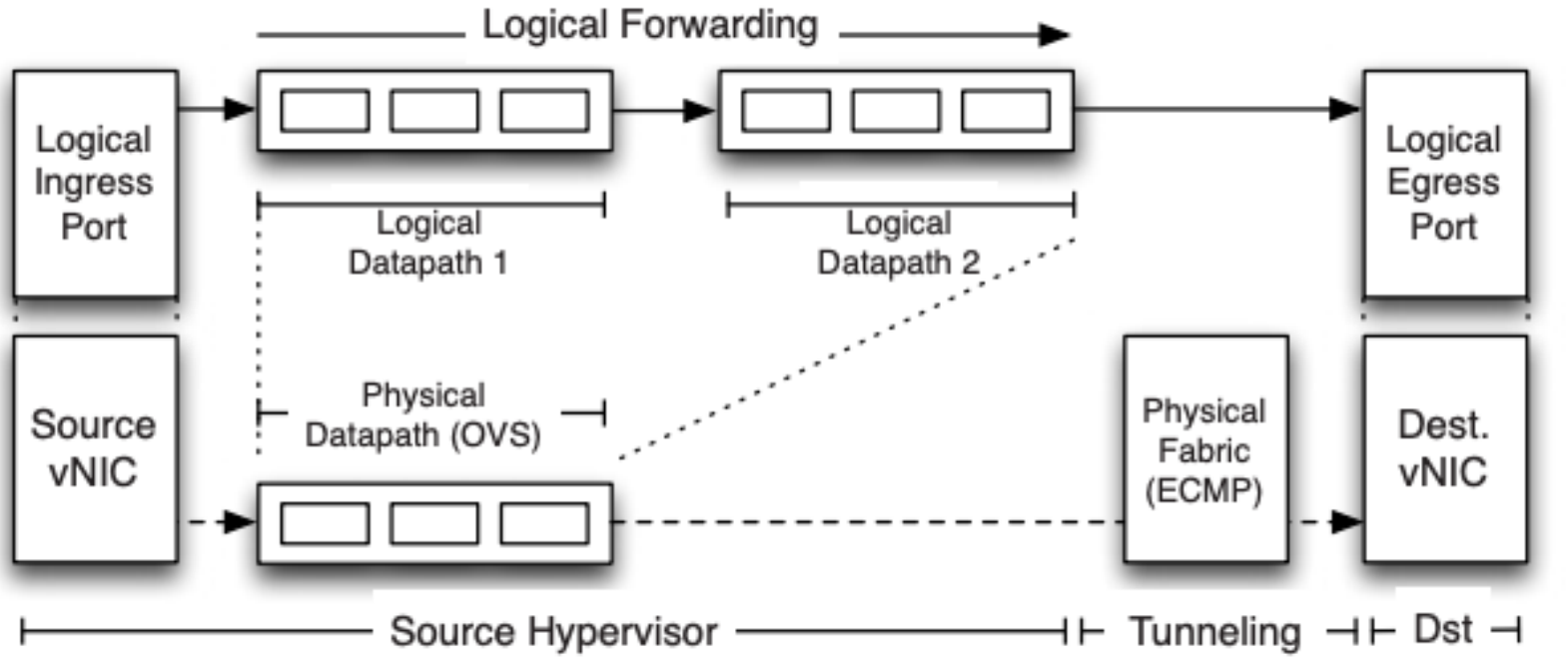


Physical Topology

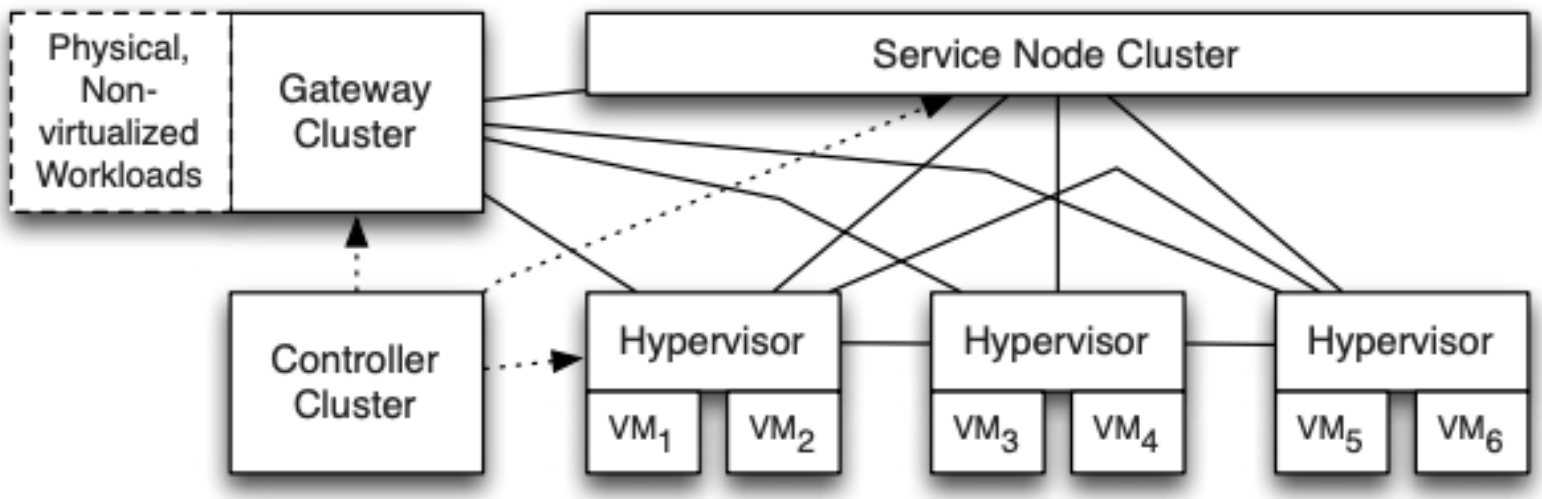


Abstraction

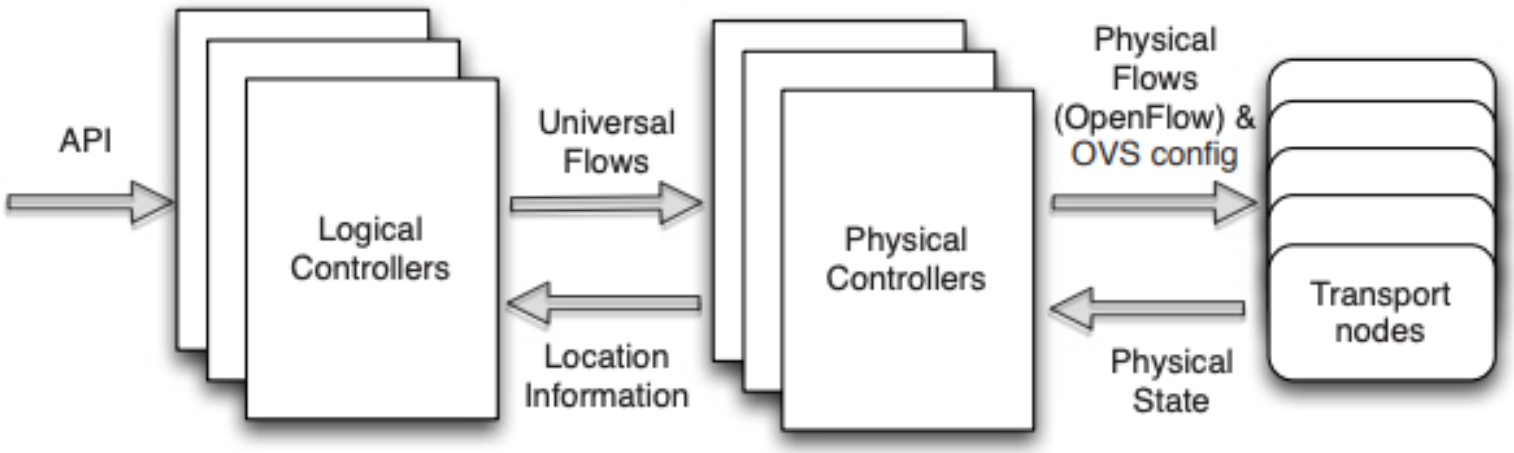
Overview


Logical Forwarding
Logical Ingress Port
Logical Datapath 1
Logical Datapath 2
Logical Egress Port
Source vNIC
Physical Datapath (OVS)
Physical Fabric (ECMP)
Dest. vNIC
Source Hypervisor
Tunneling
Dst

Architecture



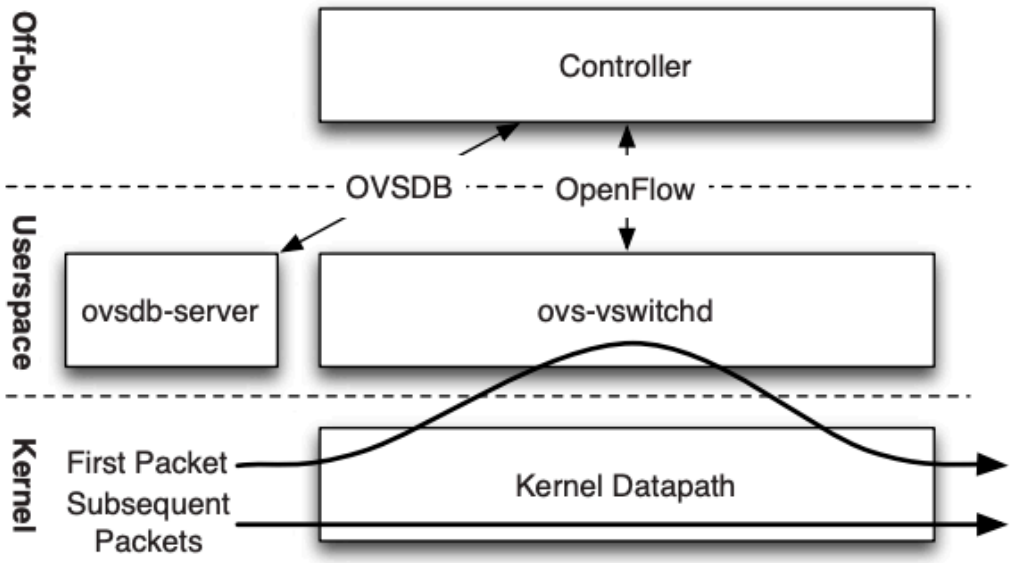
Two-layer Controller



OpenvSwitch (OVS)

- An open-source virtual switch developed at Nicira (VMWare).

- One of the most successful SDN product.



Tunneling packets between servers

- Challenge: how to support TSO/LRO?
- Solution:
  - STT: Fake (stateless) TCP header after outer IP header.
  - Issues?

Controller Scalability

- Incremental state computation.
- Multiple controller instances.
- Fault-tolerance.

Your Opinions

- Pros
  - No hardware changes.
  - Eases network configuration.
  - Use of STT.
  - Incremental state updates via nlog.
  - Distributed controller to handle scalability.

Your Opinions

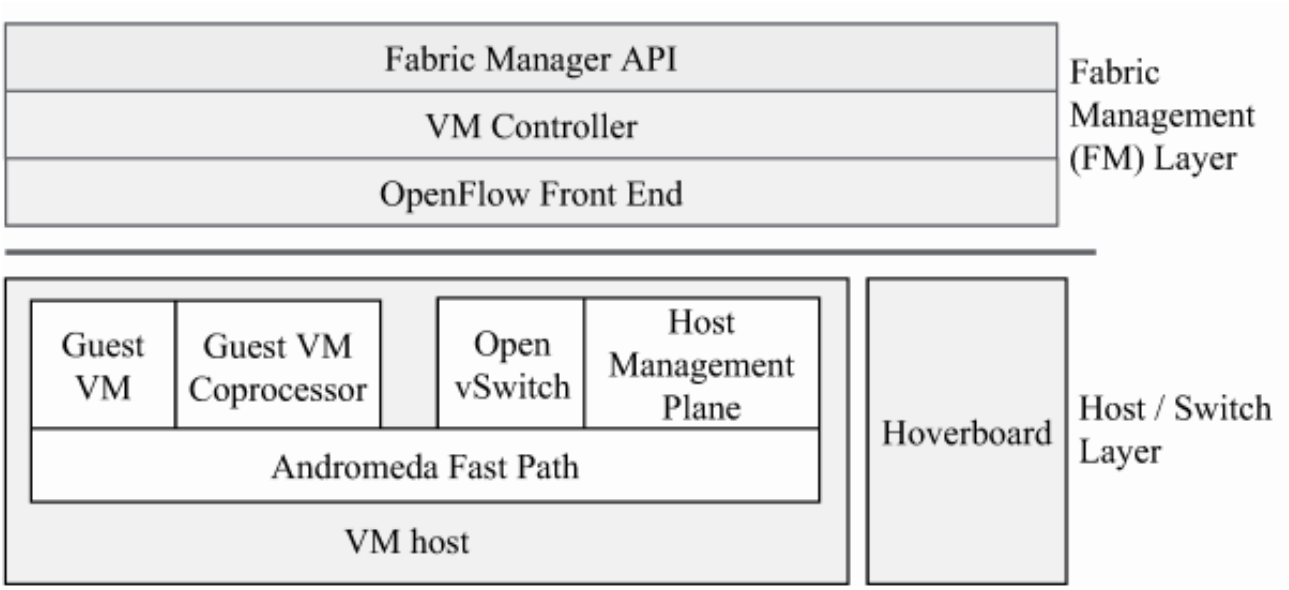
- Cons
  - Potential performance overhead of virtualization.
  - NVP can't avail itself to specialized network hardware.
  - How to provide QoS between tenants?
  - Could malicious tenants pose a security issue?
  - Long cold-start duration.

Your Opinions

- Ideas
  - Virtualize other features (load-balancers, firewalls, etc).
  - Extend to other environments beyond datacenters?
  - Can cold-start recovery be improved?
  - Scale controller at lower cost.
  - How can NVP utilize other hardware features?
  - Can one tenant impact another? (Provide isolation)

Network Virtualization at Google

- Andromeda



- SNAP and PonyExpress: Google's host-networking stack (11/05)

Network Virtualization at Microsoft

Virtual Filtering Platform (VFP)

- Overcomes certain limitations of OVS
  - Support for stateful actions.
  - Customized encapsulation/decapsulation
  - User-defined actions.

AccelNet: offload on customized hardware to make use of SR-IOV (11/10).