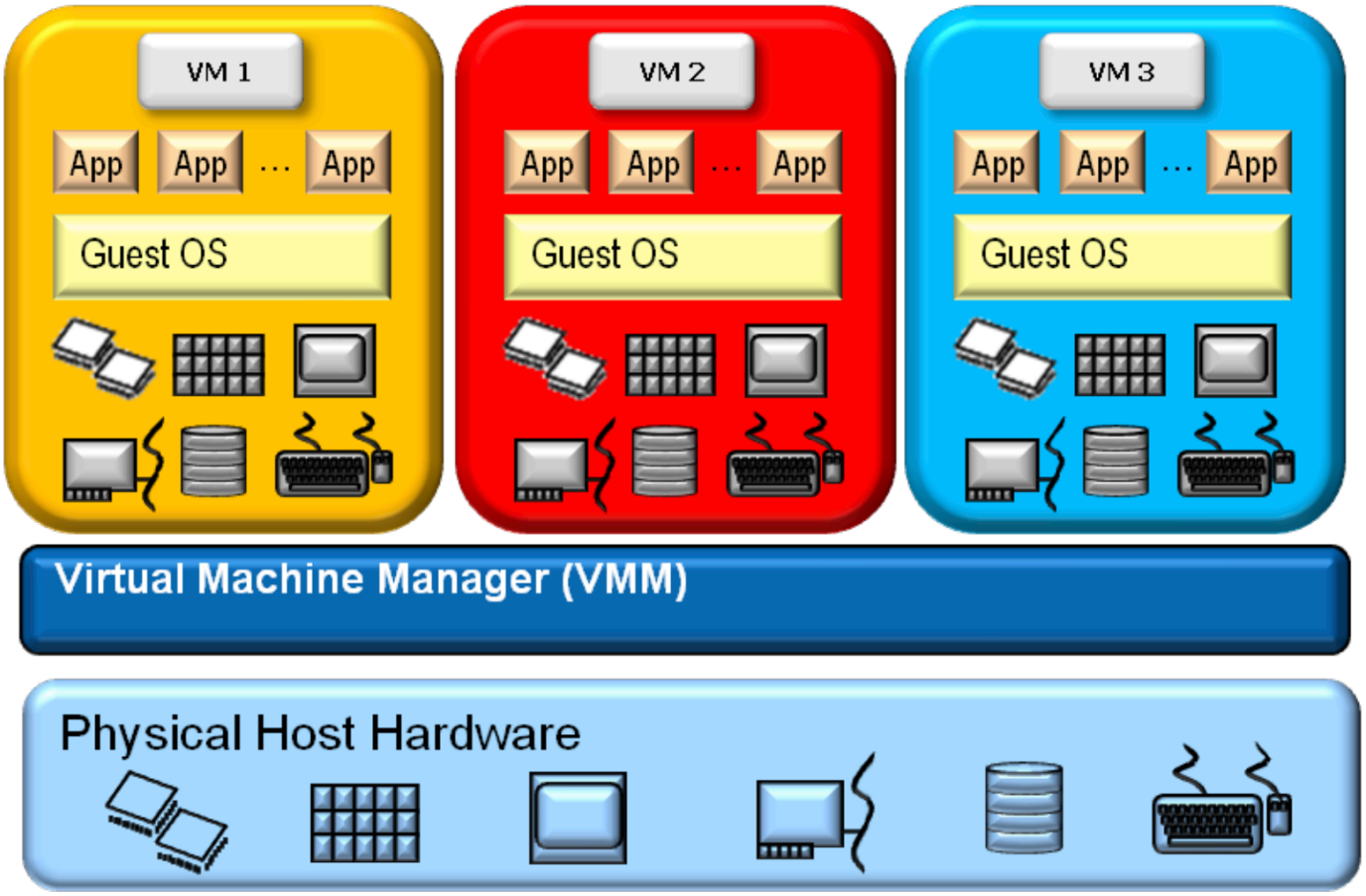
Network Virtualization and Host SDN

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 they are attached to.
  - 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 rather than individual box-by-box configuration.

Network Virtualization

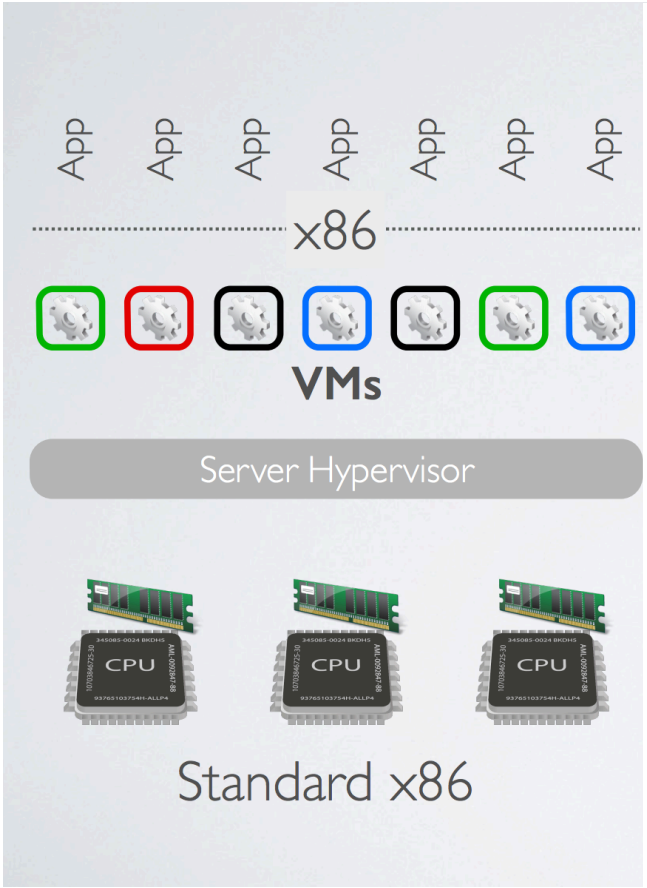
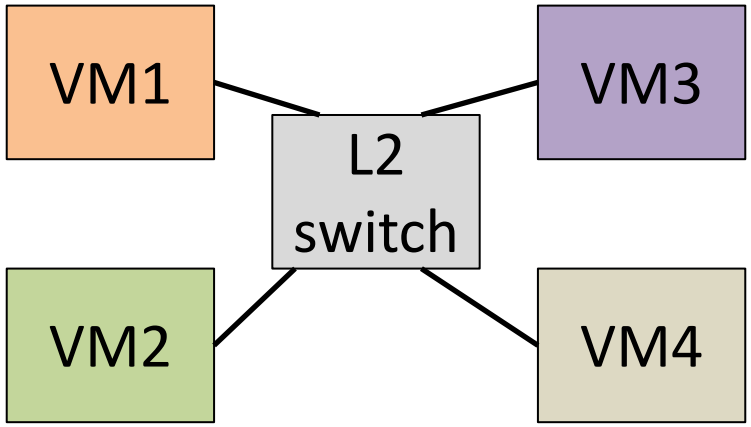
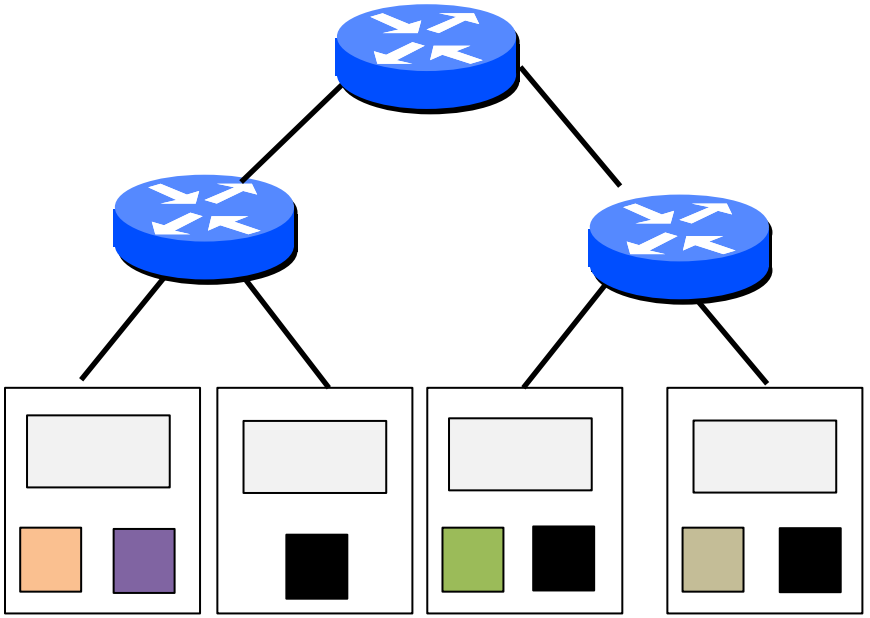


Figure from NSDI'14 talk on "Network Virtualization in Multi-tenant Datacenters"

Network Virtualization

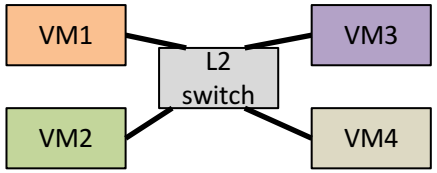


Abstraction

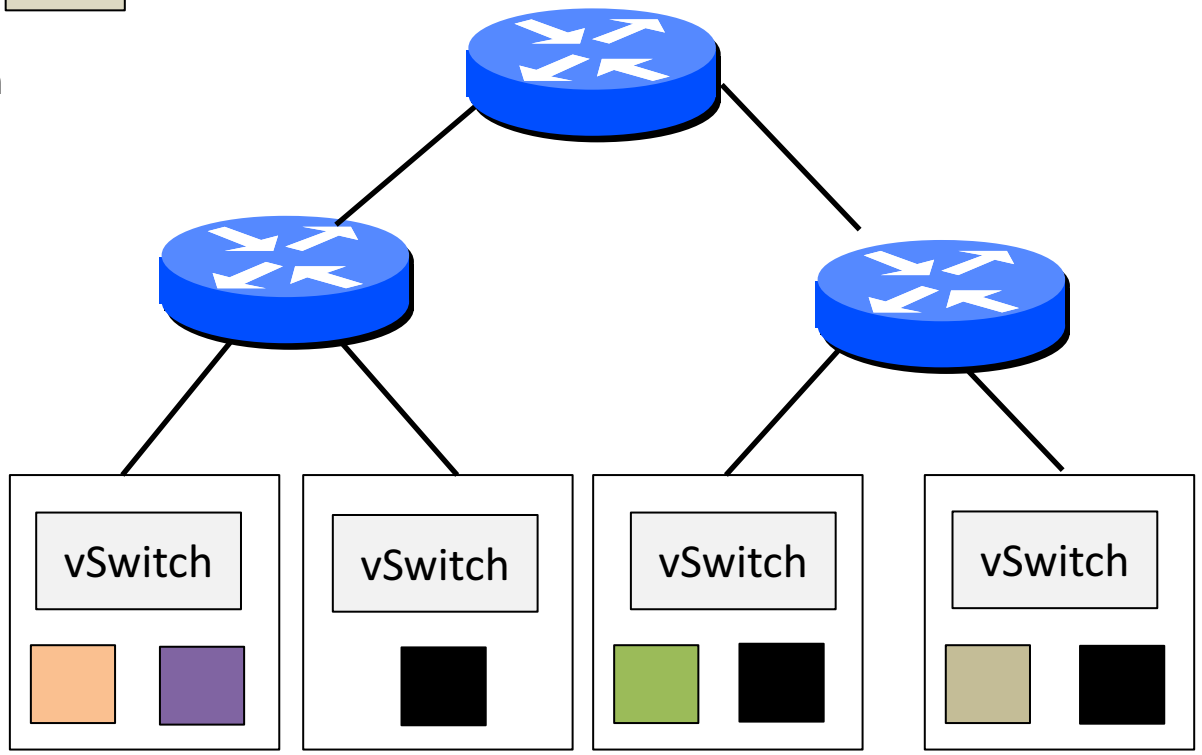


Physical Topology

vSwitches provide the abstraction



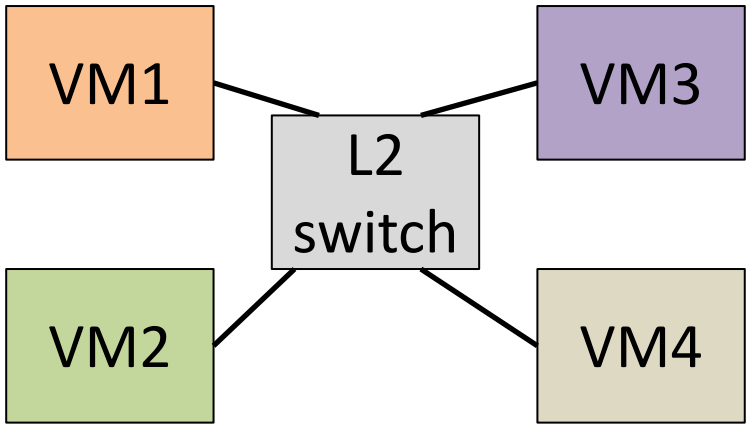
Abstraction



Physical Topology

vSwitches provide the abstraction

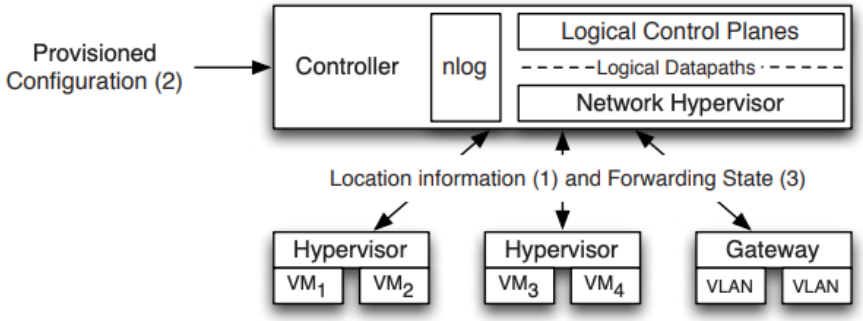
Tenants configure their policy using a centralized controller, agnostic of physical layout (control abstraction).

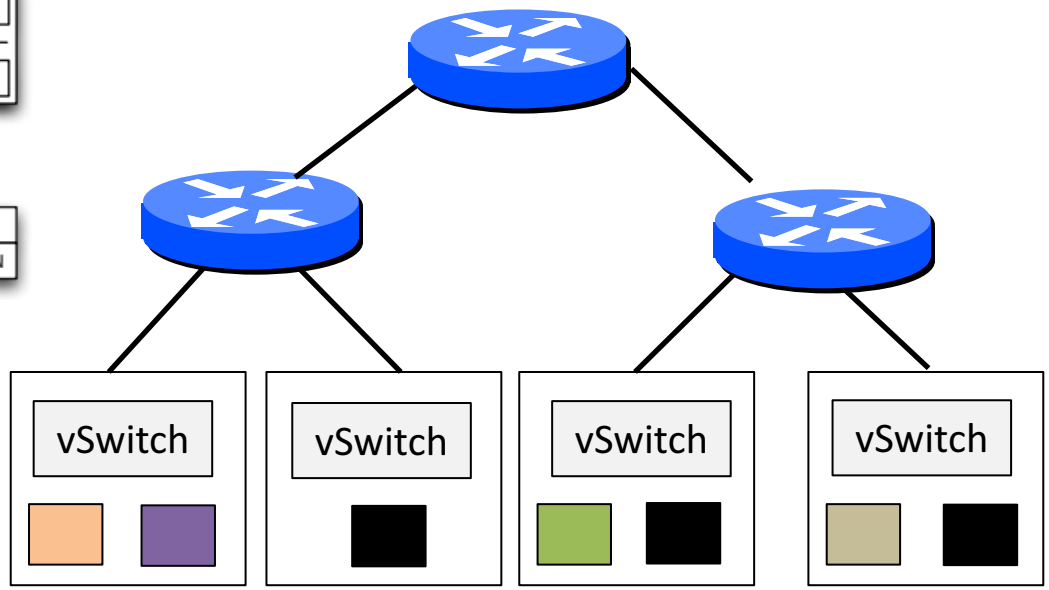


vSwitches provide the abstraction

The centralized controller cluster then translates the logical (abstract) pipeline into match-action tables executed in the vSwitch.

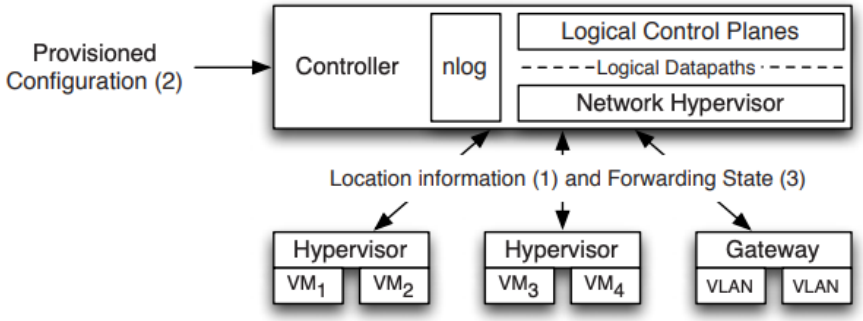
- need ways to identify which logical datapath to follow.
- takes placement of VMs into account.
- can cache the final output in the kernel datapath after first packet of a flow has been processed.

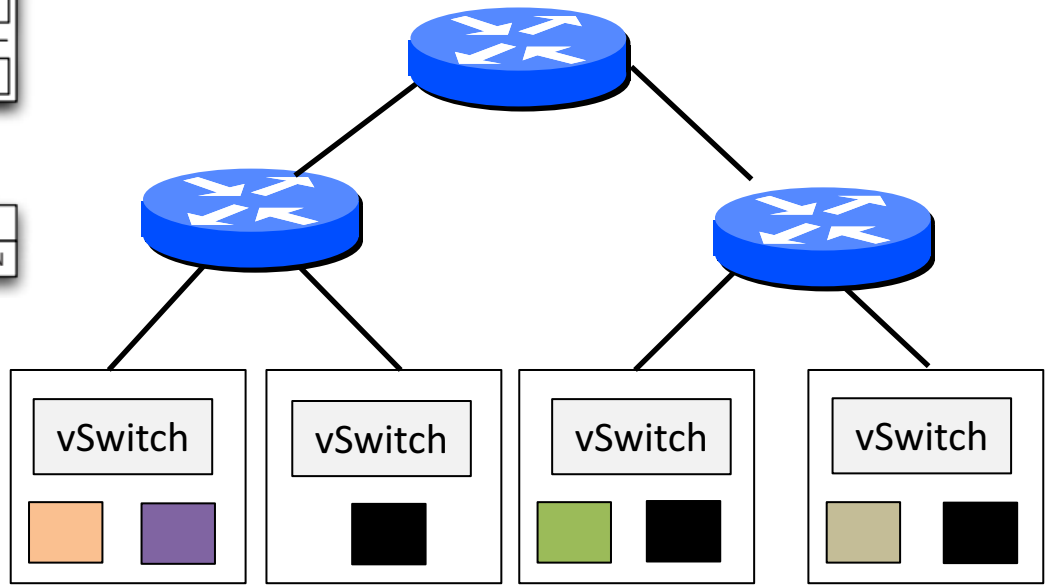




vSwitches provide the abstraction

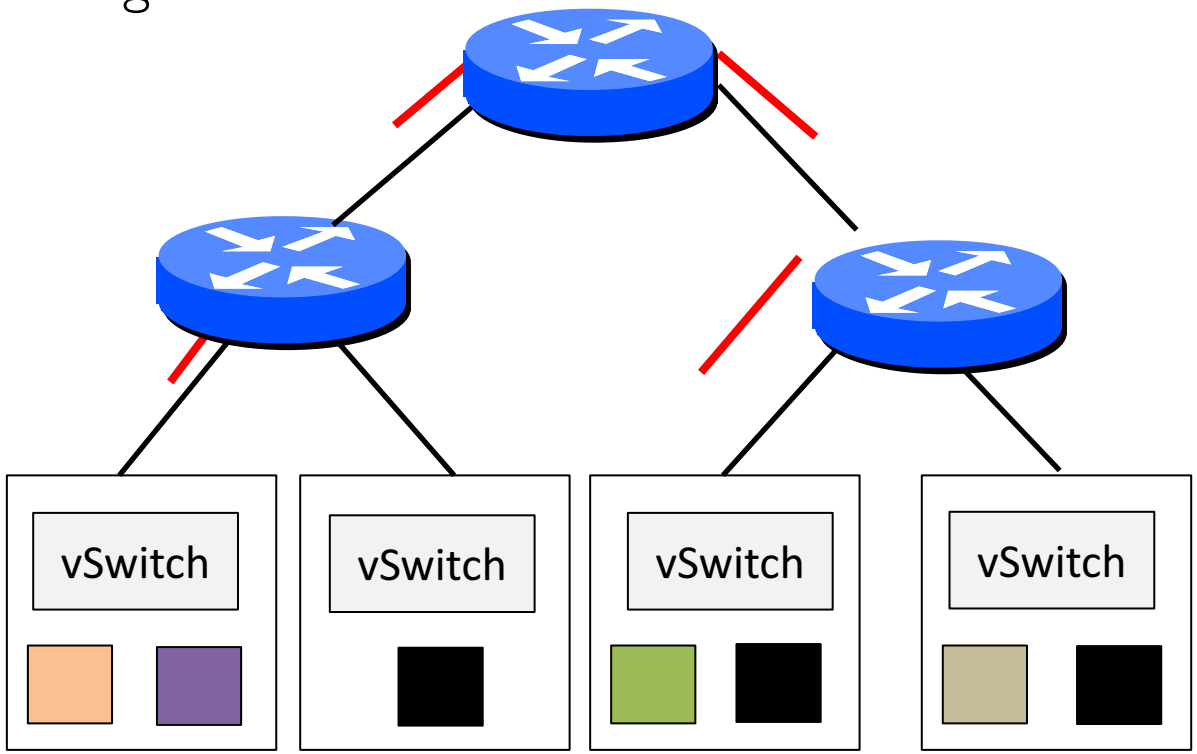
- Based on OpenFlow.
- Software vSwitches have more flexibility (and fewer constraints) than hardware switches.



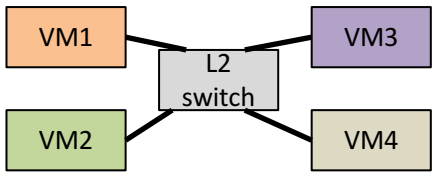


vSwitches provide the abstraction

- vSwitches transparently tunnel packets between VMs on different servers.
- Broadcast/multicast carried out by "service node overlay".
- Physical switches simply route packets from one server to another using IP-routing.



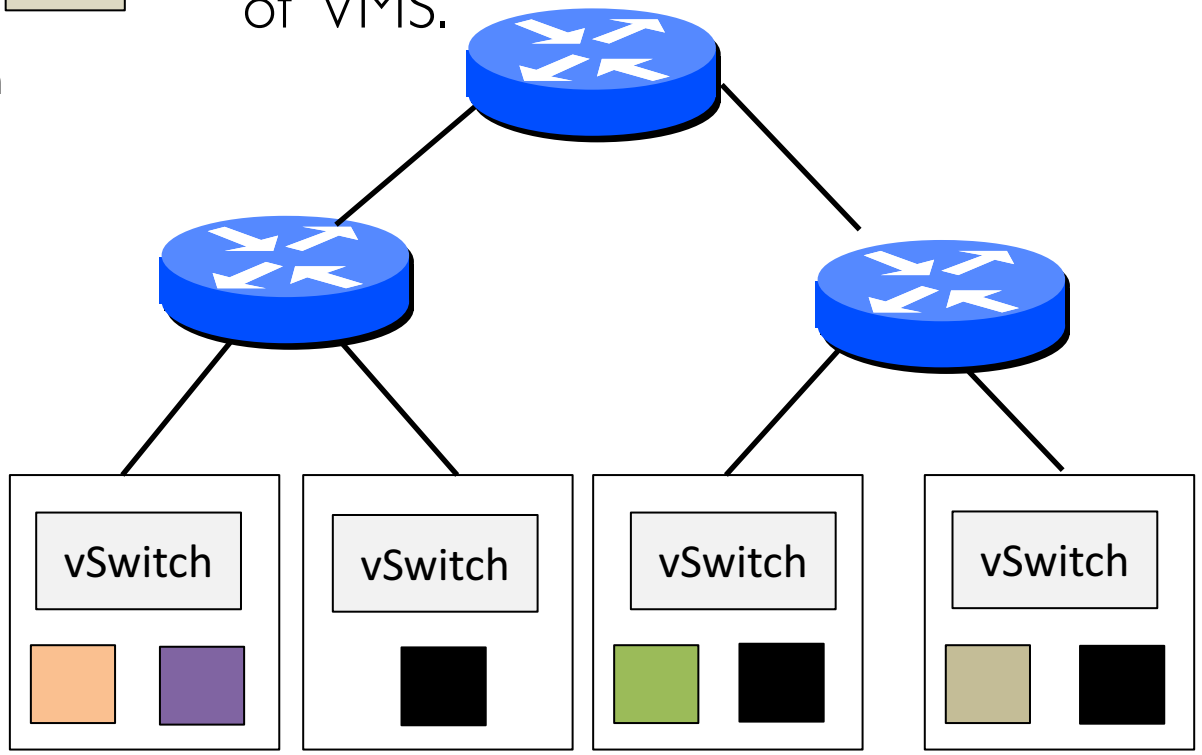
vSwitches provide the abstraction



Abstraction

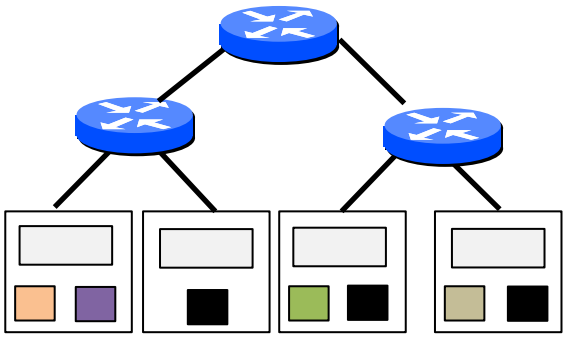
All the smartness is in the vSwitch:

- VMs behave as if they are on their own network.
- Physical network routes between servers, agnostic of VMS.

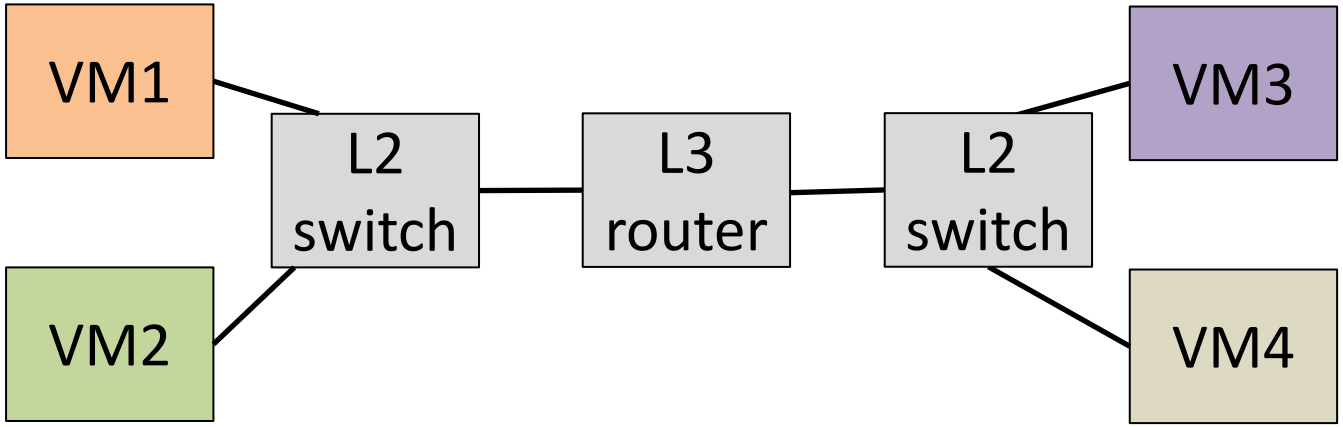


Physical Topology

More complex datapaths



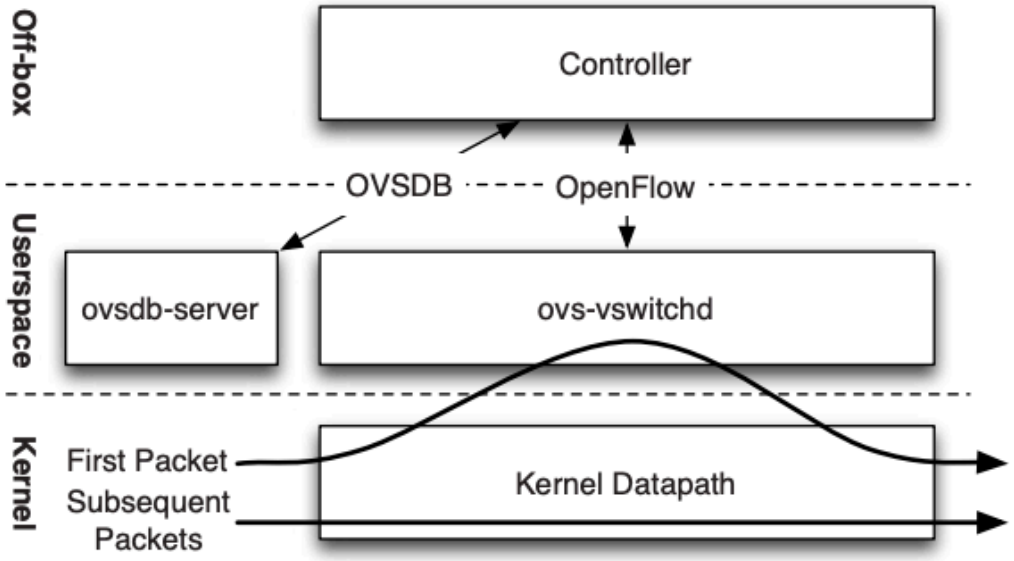
Physical Topology



Abstraction

OpenvSwitch (OVS)

- An open-source virtual switch developed at Nicira (acquired by VMWare which in turn is being acquired by Broadcom).
- One of the most successful SDN product.



OpenvSwitch 10 years later

- SIGCOMM'21
- Tight kernel user-space coupling hinders innovation and impacts performance.
- Bring most packet processing into userspace using express datapath (XDP).
- An alternative was to use OVS over DPDK, but that was incompatible with commonly used tools and systems.

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.

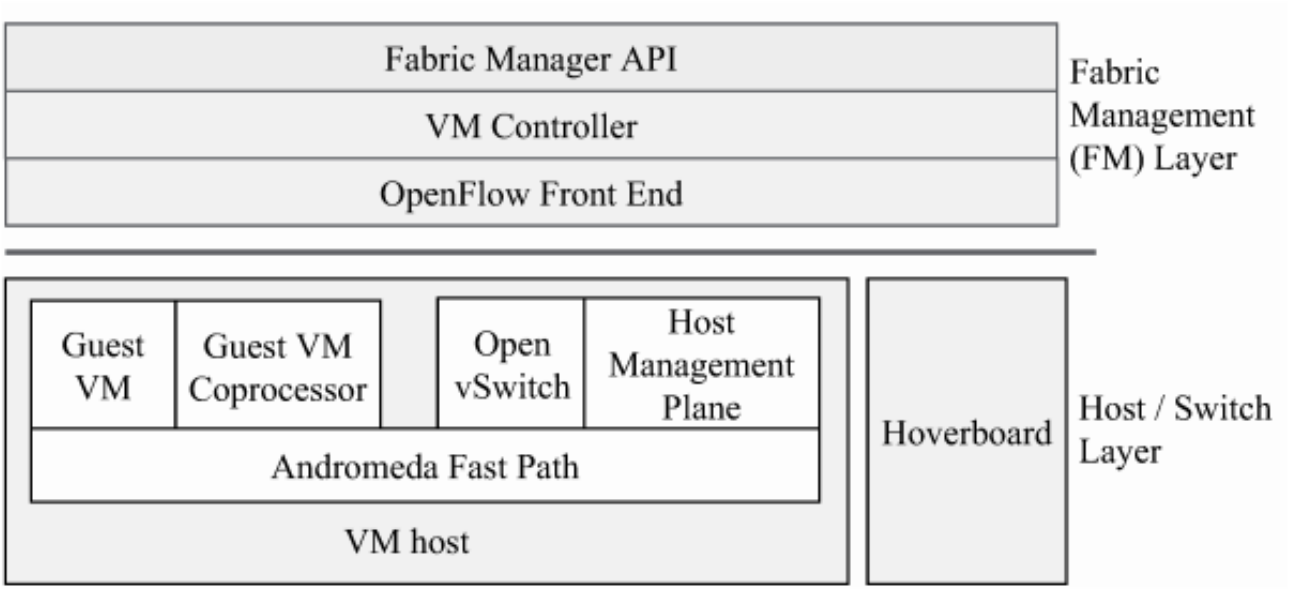
Your thoughts?

- What did you like about the paper?

- What are its limitations?

Network Virtualization at Google

- Andromeda (NSDI'18)



- SNAP and PonyExpress (SOSP'19): Google's host-networking stack

Network Virtualization at Microsoft

Virtual Filtering Platform (VFP), NSDI'17

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

AccelNet (NSDI'18): offload on customized hardware to make use of SR-IOV.