# Network Virtualization

Ben Pfaff

Nicira Networks, Inc.

# Preview

- Data Centers
- Problems: Isolation, Connectivity
- Solution: Network Virtualization
- Network Tunnels
- A Network Virtualization Architecture
- Open vSwitch Design
- Questions?

# Data Centers

Front of a rack

Rear of a rack

A data center has many racks.

A rack has 20-40 servers.

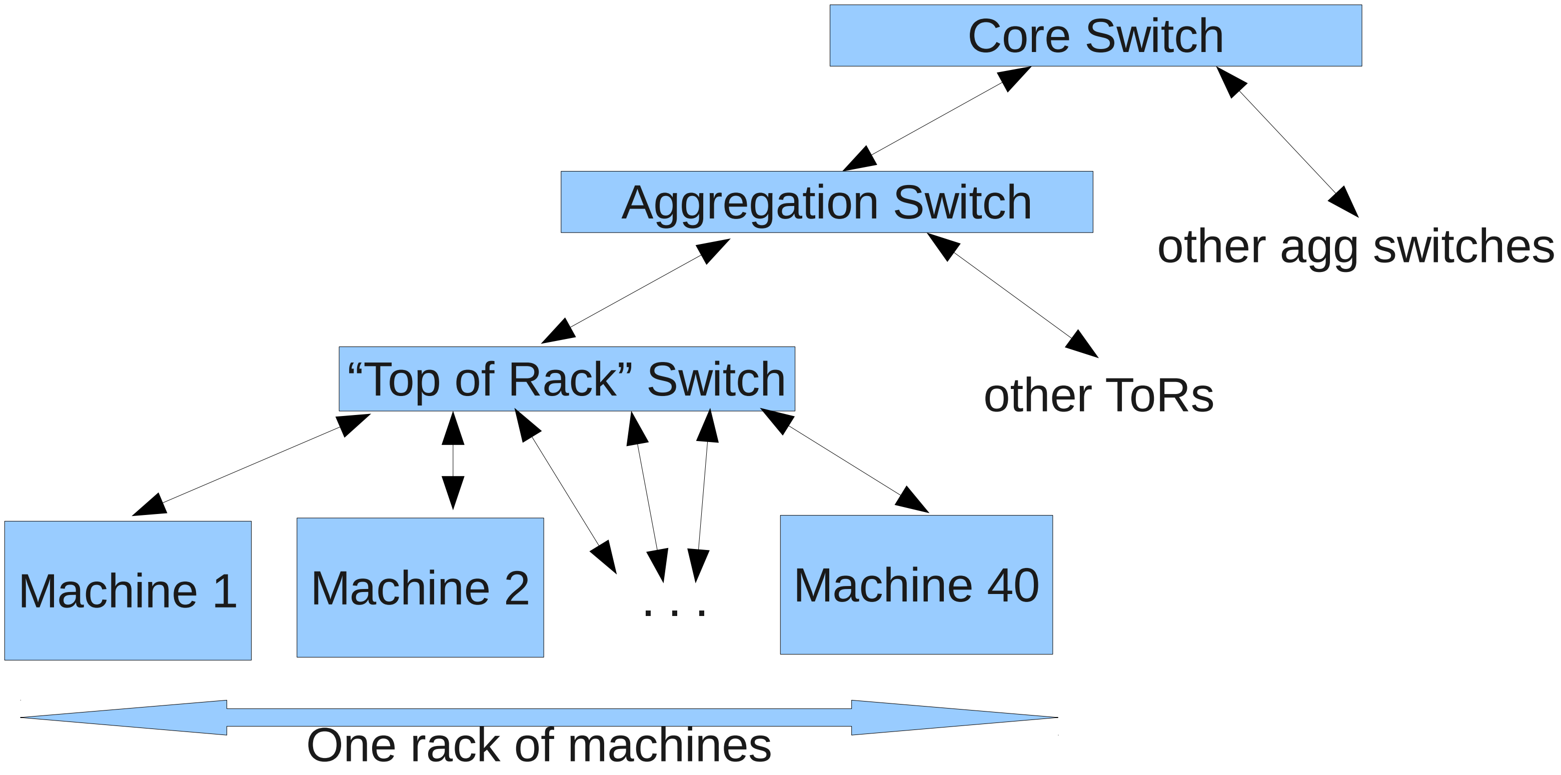
The servers in a rack connect to a 48-port “top of rack” (ToR) switch.

Data centers buy the cheapest ToR switches they can find.  They are pretty dumb devices.

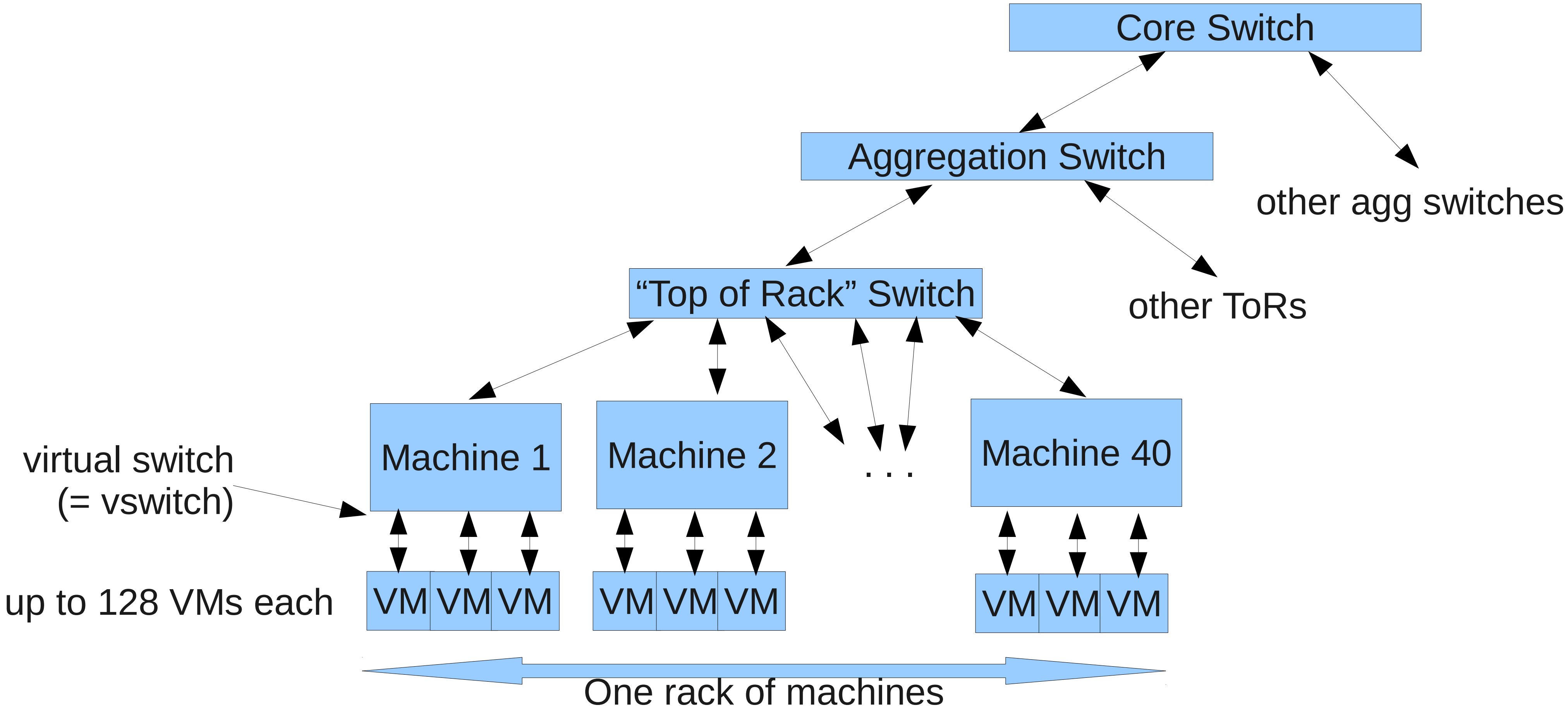
Data centers do not rewire their networks without a really good reason.

“Top of Rack” switch

# Data Center Network Design before VMs



# Data Center Network Design with VMs



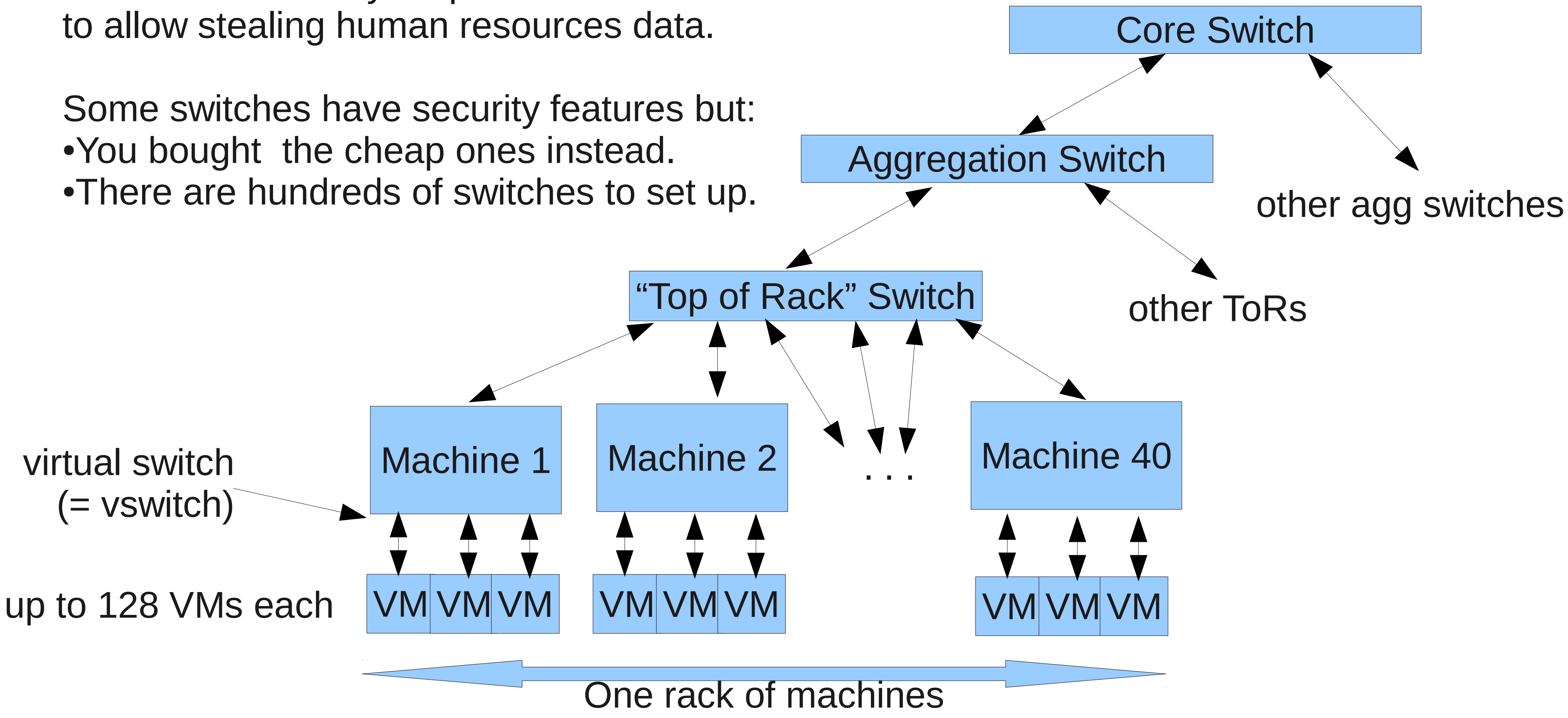
# Problem: Isolation

All VMs can talk to each other by default.

You don't want someone in engineering screwing up the finance network. You don't want a break-in to your production website to allow stealing human resources data.

Some switches have security features but:

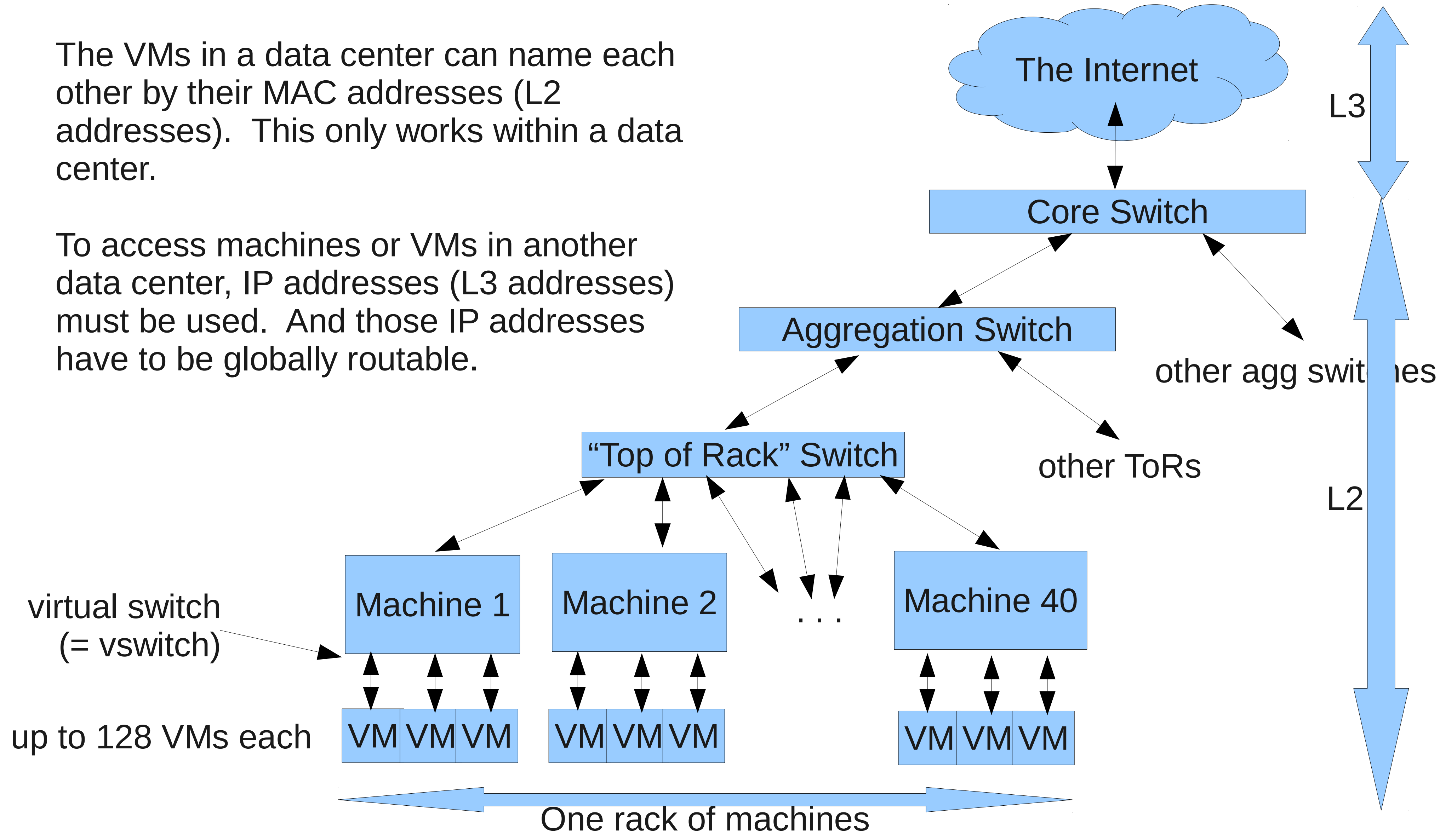
- You bought the cheap ones instead.
- There are hundreds of switches to set up.



# Problem: Connectivity

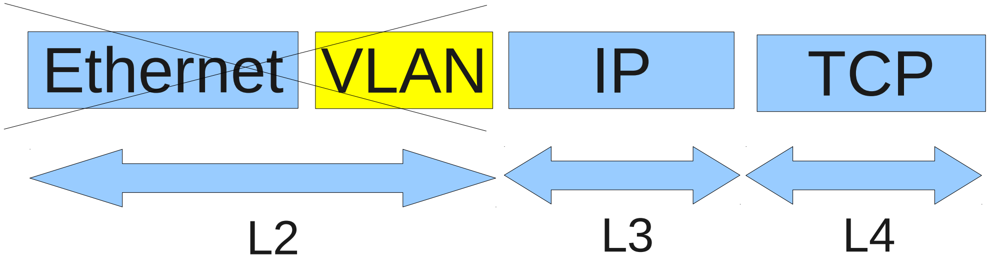
The VMs in a data center can name each other by their MAC addresses (L2 addresses). This only works within a data center.

To access machines or VMs in another data center, IP addresses (L3 addresses) must be used. And those IP addresses have to be globally routable.



# Non-Solution: VLANs

A VLAN partitions a physical Ethernet network into isolated virtual Ethernet networks:



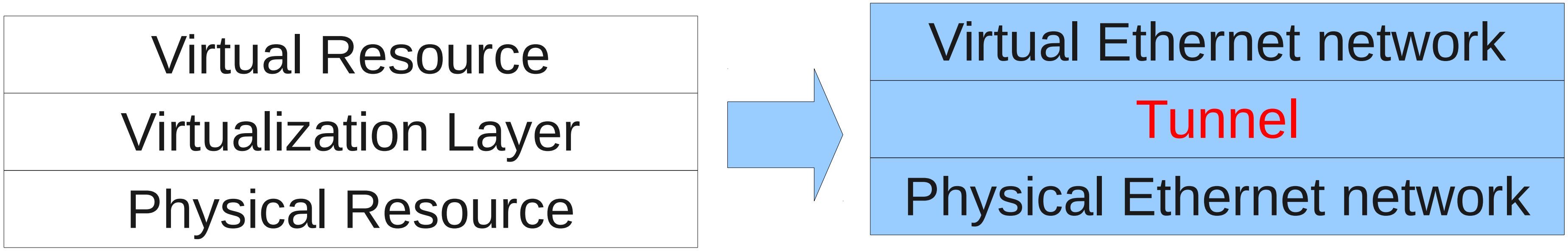
The Internet is an L3 network.  When a packet crosses the Internet, it loses all its L2 headers, including the VLAN tag.  You lose all the isolation when your traffic crosses the Internet.

Other problems: limited number, static allocation.

# Solution: Network Virtualization

Virtualization Layering

Network Virtualization



Virtual Resource

Virtualization Layer

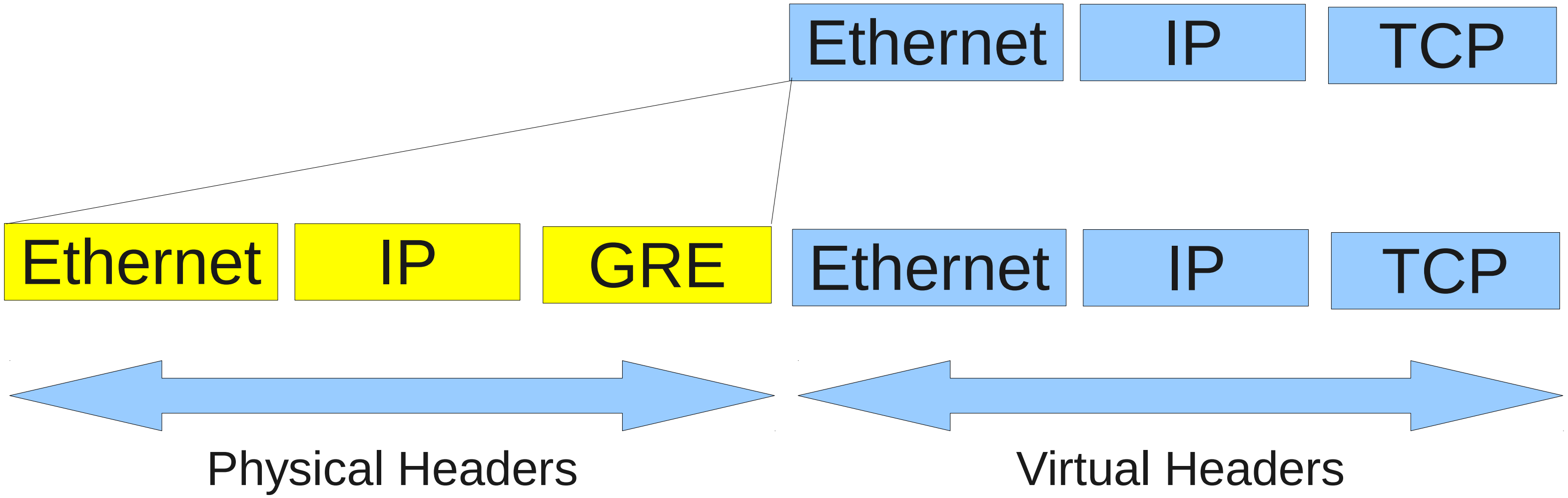
Physical Resource

Virtual Ethernet network

Tunnel

Physical Ethernet network

## Tunneling: Separating Virtual and Physical Network



Ethernet IP TCP

Ethernet IP GRE Ethernet IP TCP

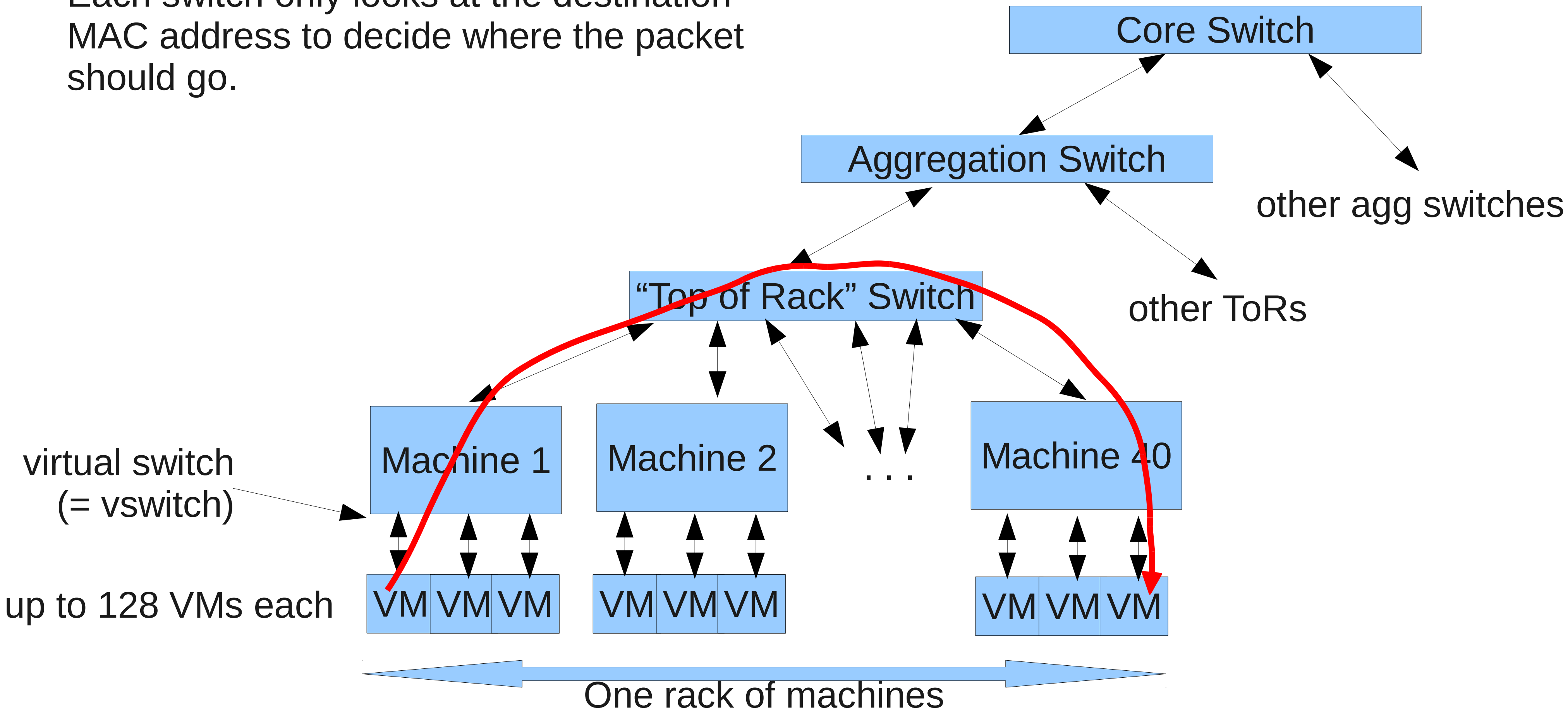
Physical Headers

Virtual Headers

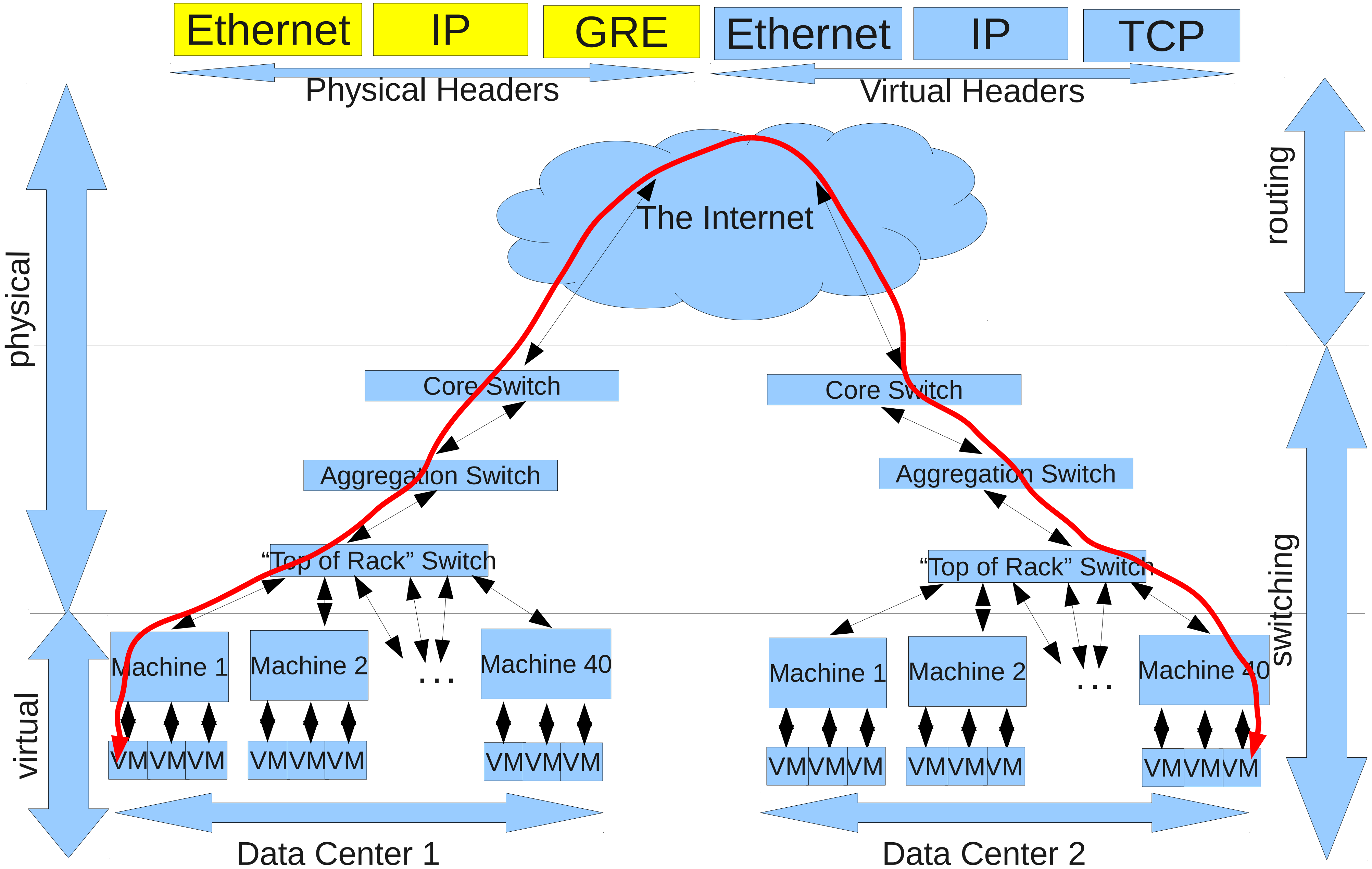
# Path of a Packet (No Tunnel)

A packet from one VM to another passes through a number of switches along the way.

Each switch only looks at the destination MAC address to decide where the packet should go.



# Path of a Packet (Via Tunnel)



Ethernet IP GRE

Physical Headers

Ethernet IP TCP

Virtual Headers

physical

virtual

routing

switching

The Internet

Core Switch

Aggregation Switch

“Top of Rack” Switch

Machine 1

Machine 2

. . .

Machine 40

VM VM VM

Data Center 1

Core Switch

Aggregation Switch

“Top of Rack” Switch

Machine 1

Machine 2

. . .

Machine 40

VM VM VM

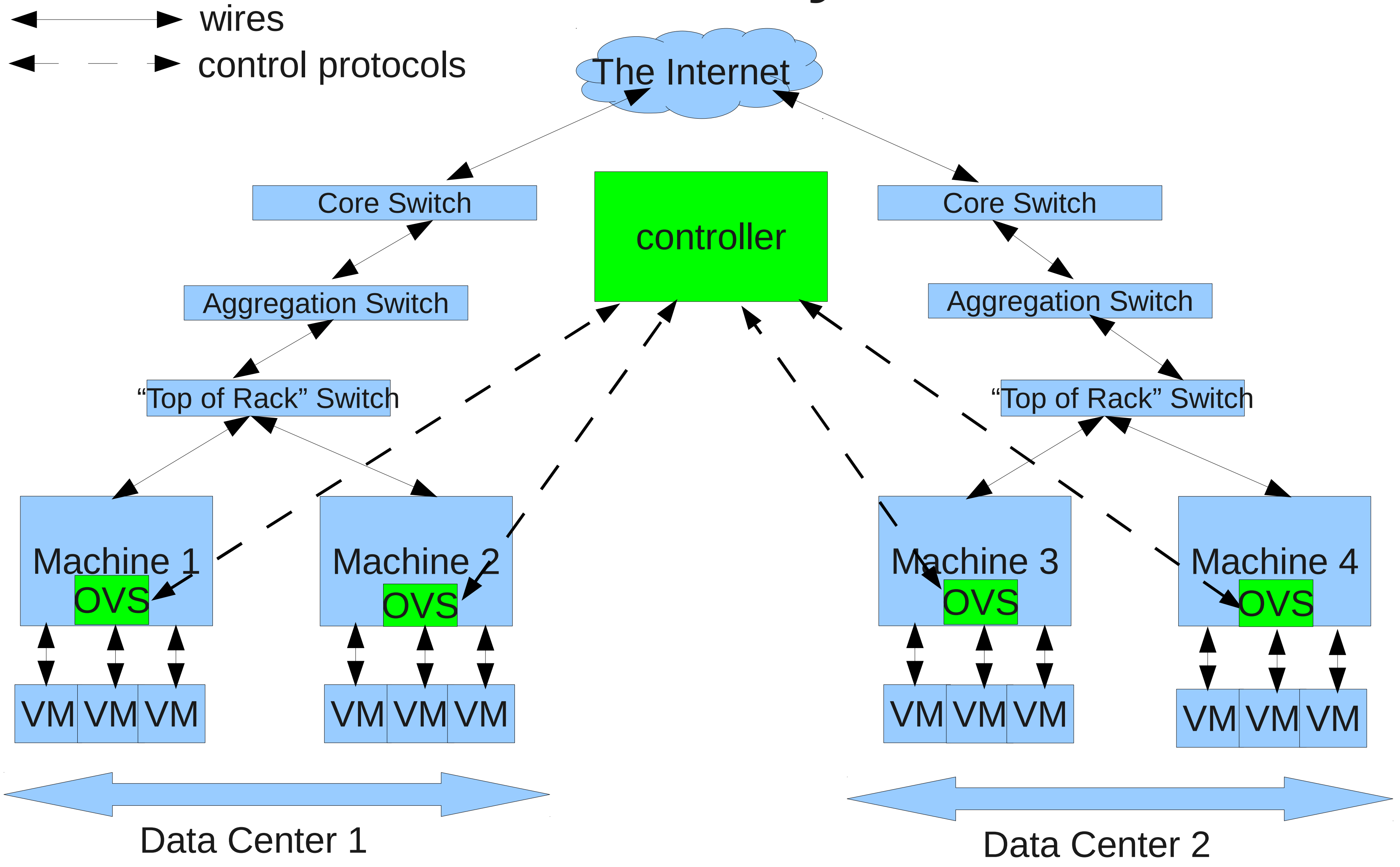
Data Center 2

# Challenges

- Setting up the tunnels:
  - Initially.
  - After VM startup/shutdown
  - After VM migration
- Handling network failures
- Monitoring, administration
- Administration

⇒ Use a central controller to set up the tunnels.

# A Network Virtualization Distributed System



# Controller Duties

- Monitor:
  - Physical network
  - VM locations, states
- Control:
  - Tunnel setup
  - All packets on virtual and physical network
  - Virtual/physical mapping
- Tells OVS running everywhere else what to do

# Open vSwitch

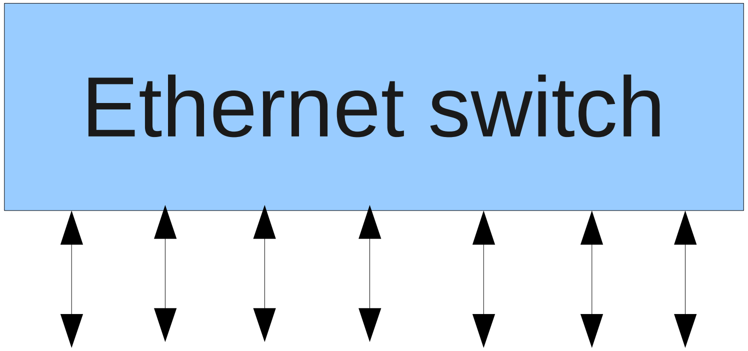
- Ethernet switch implemented in software
- Can be remotely controlled
- Tunnels (GRE and others)
- Integrates with VMMs, e.g. XenServer, KVM
- Free and open source

openvswitch.org

# Open vSwitch: OVSDB protocol

- Slow-moving state:
  - VM placement (via VMM integration)
  - Tunnel setup
- Buzzwords:
  - Lightweight
  - Transactional
  - Not SQL
  - Persistent

# Open vSwitch: OpenFlow protocol

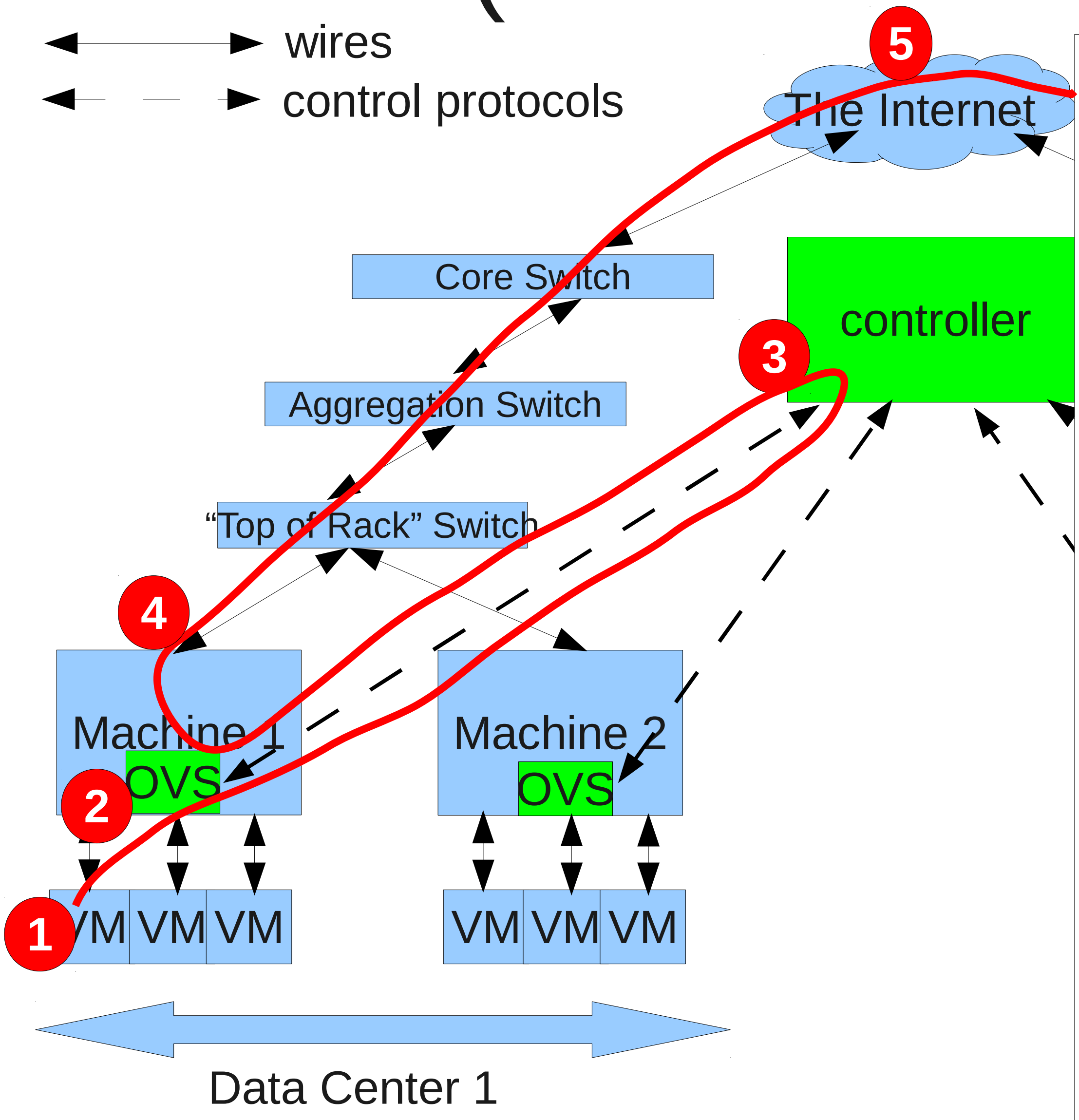


Flow table = ordered list of “if-then” rules:

“If this packet comes from VM A and going to VM B, then send it out via tunnel 42.”

(No rule: send to controller.)

# OpenFlow in the Data Center (One Possibility)

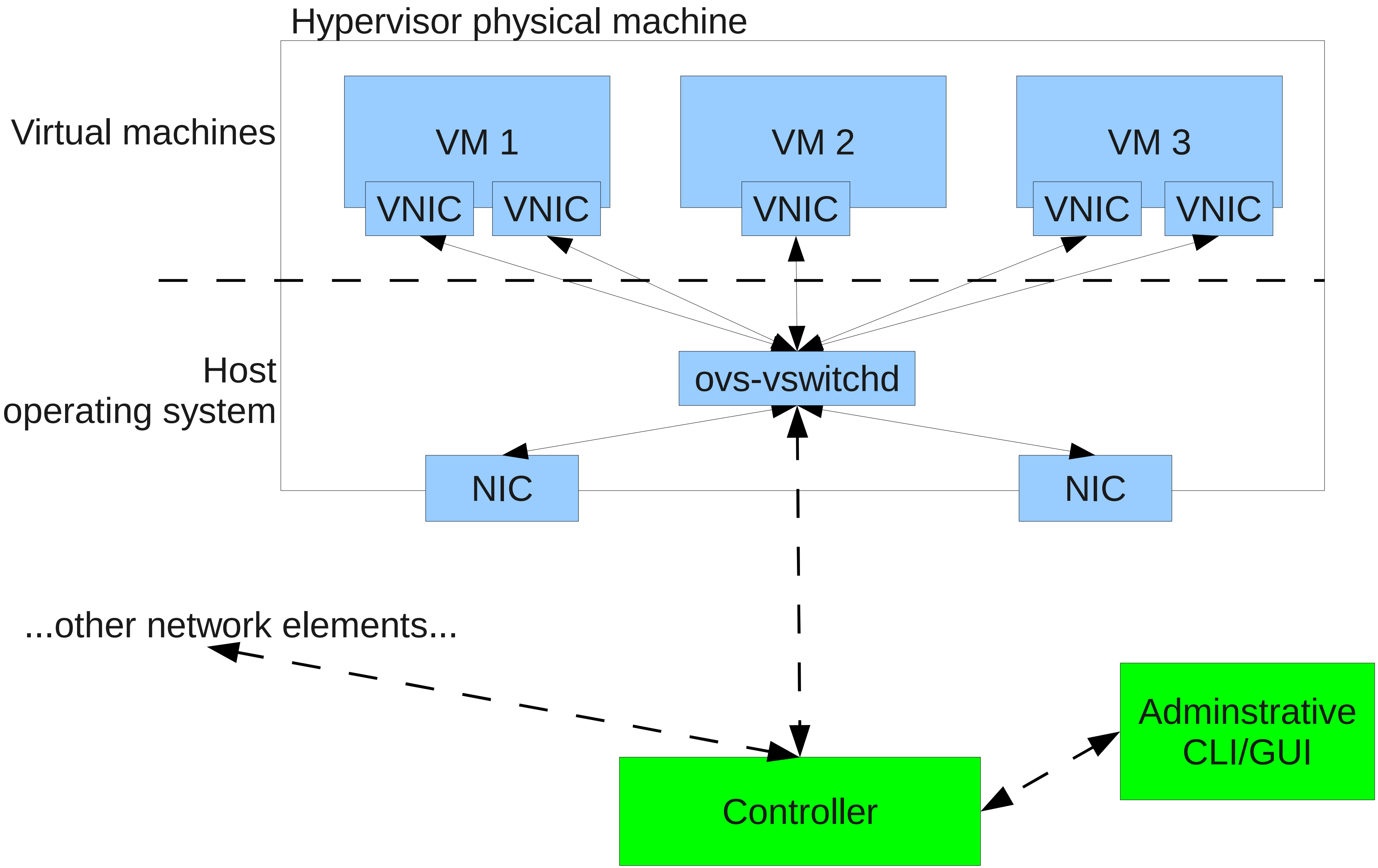


1. VM sends packet.
2. Open vSwitch checks flow table – no match. Sends packet to controller.
3. Controller tells OVS to set up a tunnel to the destination and send the packet on that tunnel.
4. OVS sends packet on the new tunnel.
5. Normal switching and routing carry the packet to its destination in the usual way.

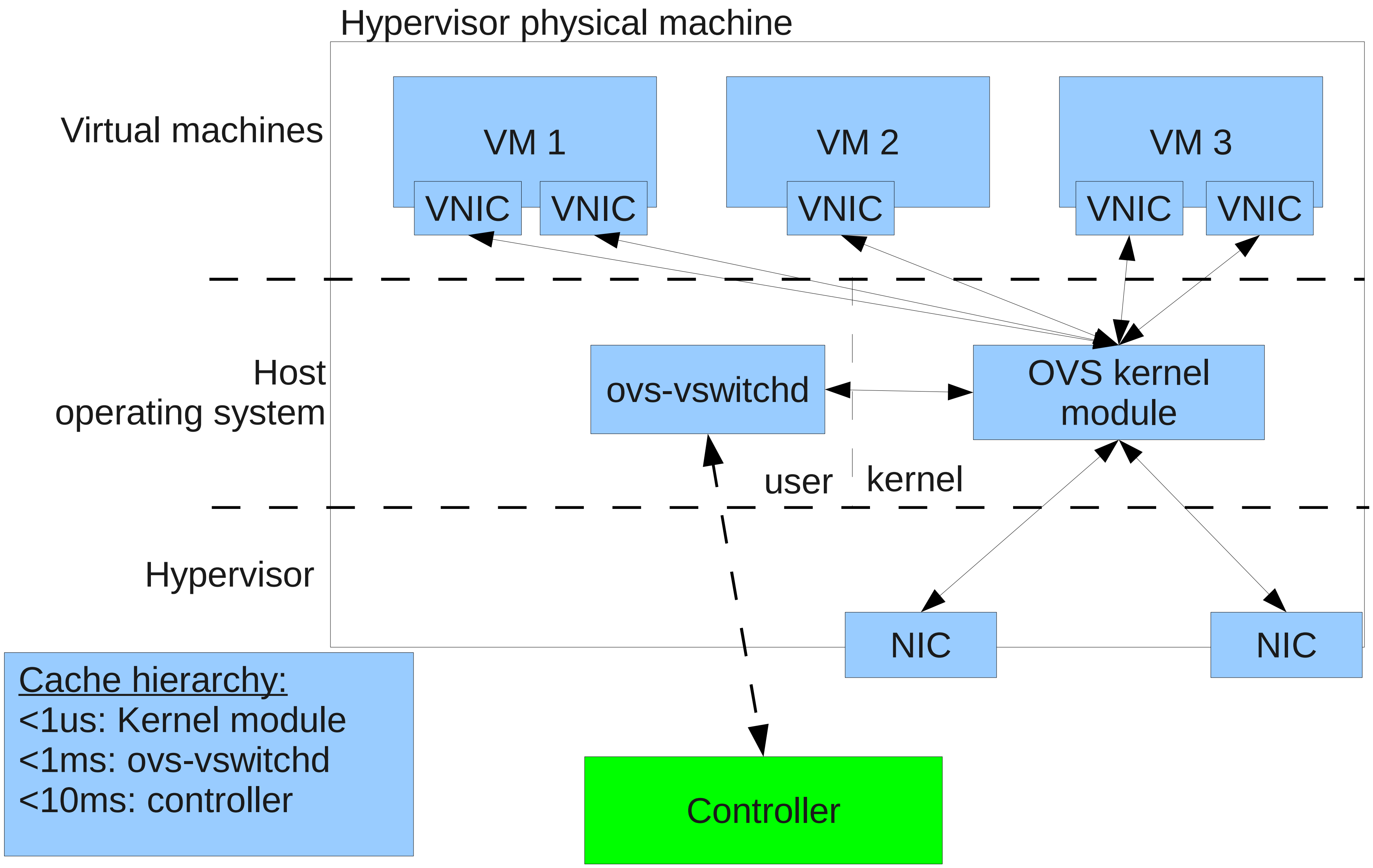
The same process repeats on the other end to send the reply back.

This is done at most on a per-”flow” basis, and other optimizations keep it from happening too frequently.

# Open vSwitch: Design Overview



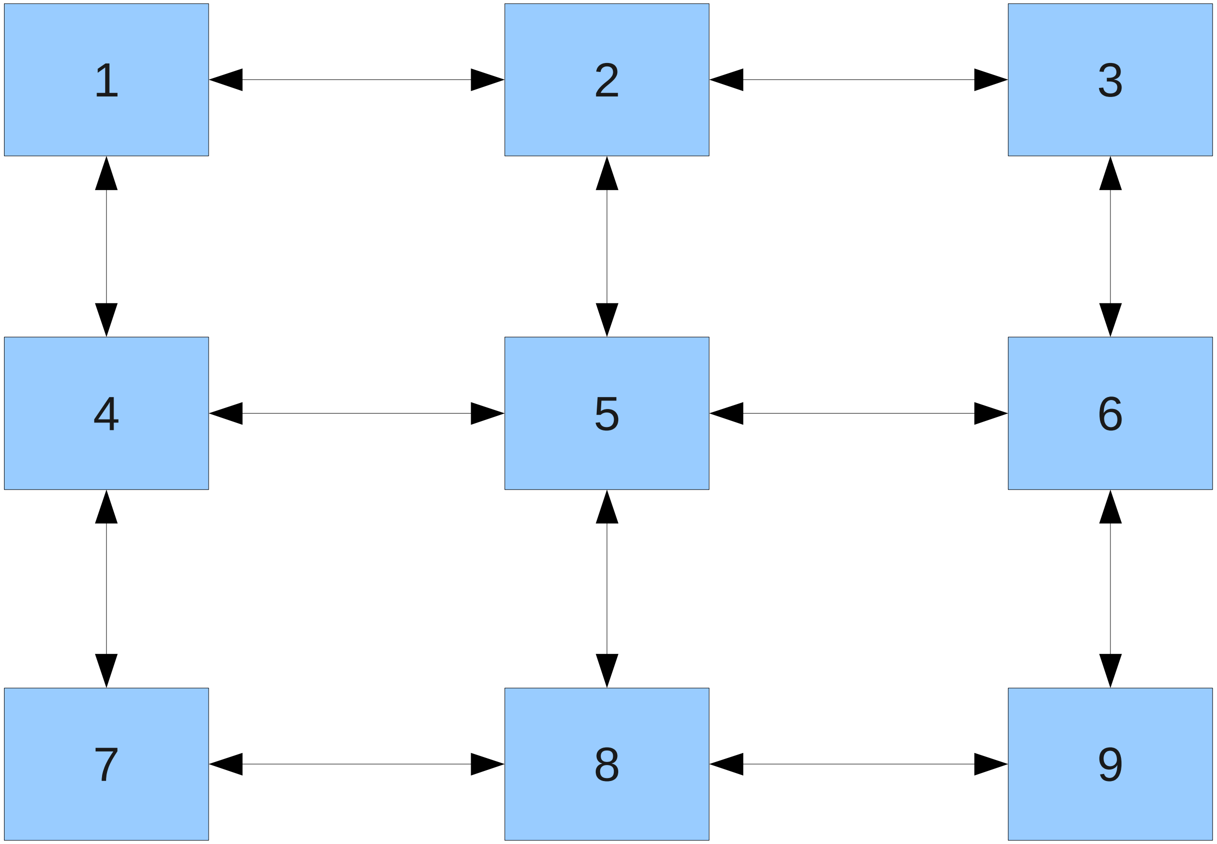
# Open vSwitch: Design Details



Cache hierarchy:
<1us: Kernel module
<1ms: ovs-vswitchd
<10ms: controller

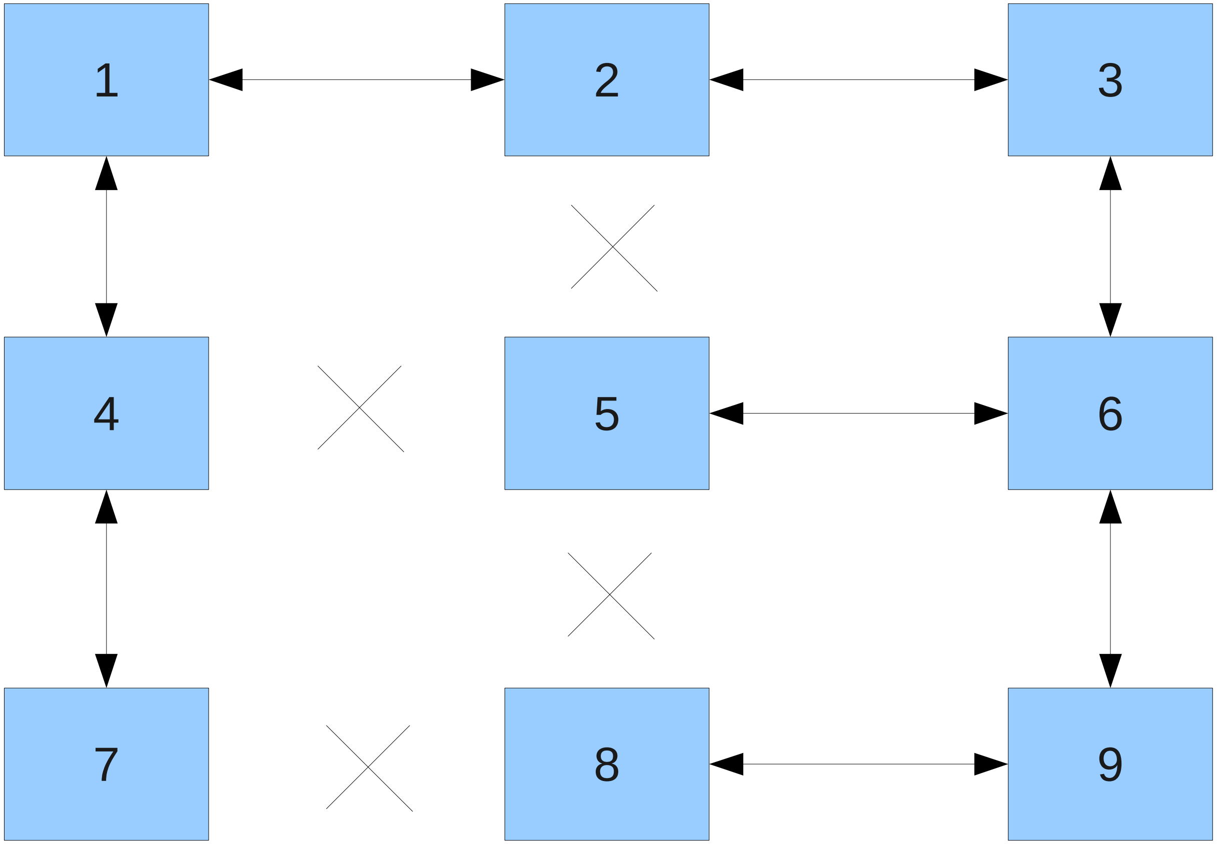
# OpenFlow: Another Use

Redundant Wiring
Between Switches:



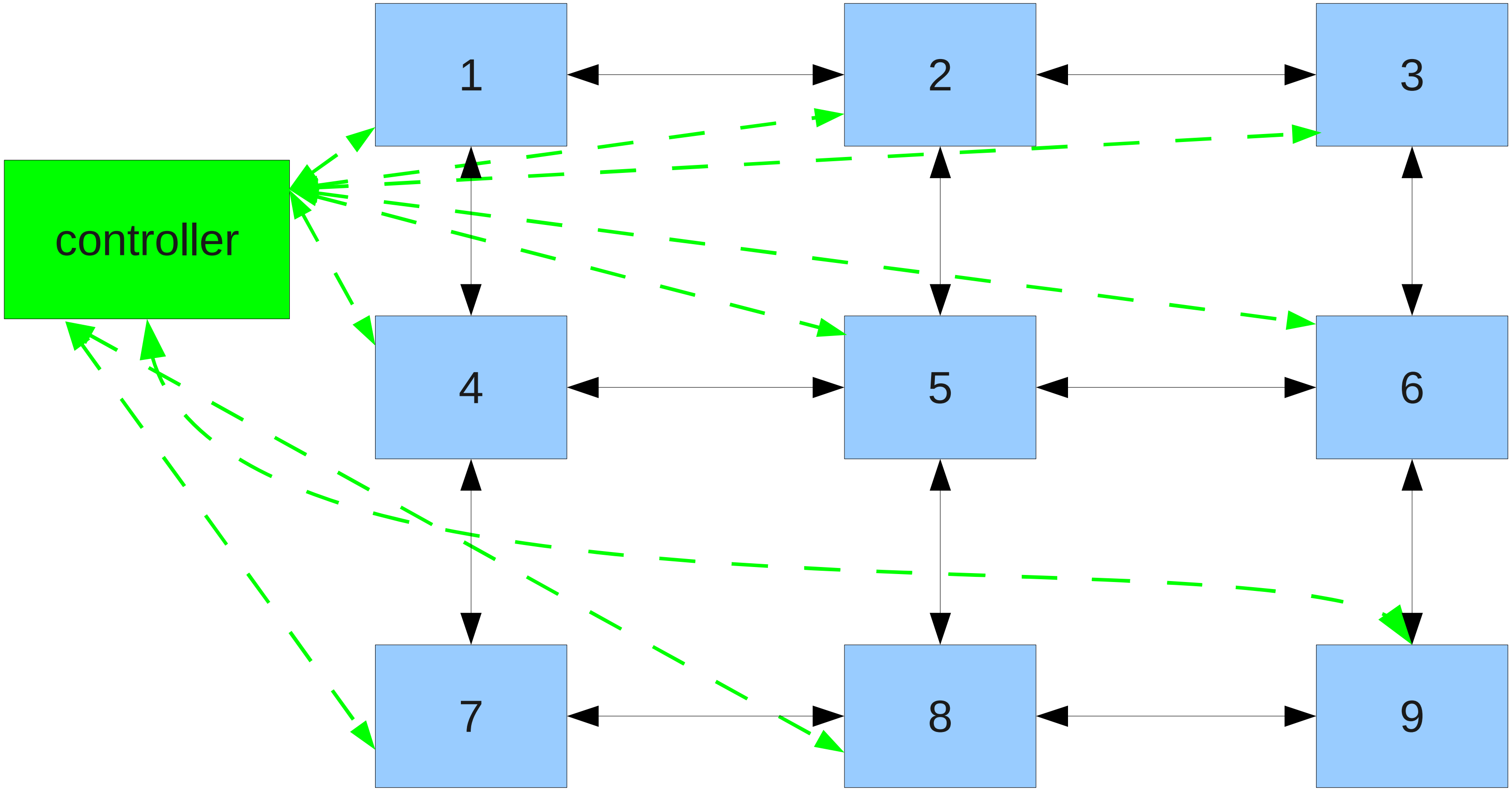
# OpenFlow: Another Use

Effective topology with
normal L2 switching:



# OpenFlow: Another Use

L2 routing managed by controller:



(Requires all switches to support OpenFlow)

# Conclusion

- Companies spread VMs across data centers.
- Ordinary networking exposes differences between VMs in the same data center and those in different data centers.
- Tunnels can hide the differences.
- A controller and OpenFlow switches at the edge of the network can set up and maintain the tunnels.

# Questions?