Network Virtualization:
Delivering on the Promises of SDN

Bruce Davie, Principal Engineer

vmware®

What does SDN promise?

- **The original* SDN promise was:**
  - Enable rapid innovation in networking
  - Enable new forms of network control

* "SDN" was coined as an umbrella term to encompass separation of control and data plane (including but not limited to OpenFlow)

vmware®

March/April 2009

TR10: Software-Defined Networking

Nick McKeown believes that remotely controlling network hardware with software can bring t

Kate Greene

More recent SDN claims

- **Vendor choice**
  - Theory: hardware and software from different vendors, more diversity in both
  - This has yet to play out

- **Simplified Programmability**
  - Theory: move away from all those complex distributed algorithms of traditional networking (see next slide)

- **Enable application-level control/programming of network**
  - Remember Integrated Services & RSVP? Just like that, but awesomer!

- **Simplified operations, provisioning, etc.**
  - Theory: SDN enables centralized control, leading to simplification

vmware®

SDN provides:

- Saner dev environment: server vs. embedded wimpy cpu
- Distributed coordination primitives that scale

protocols

coordination primitives (e.g. locking)

control

control

control

control

control

Traditional

SDN
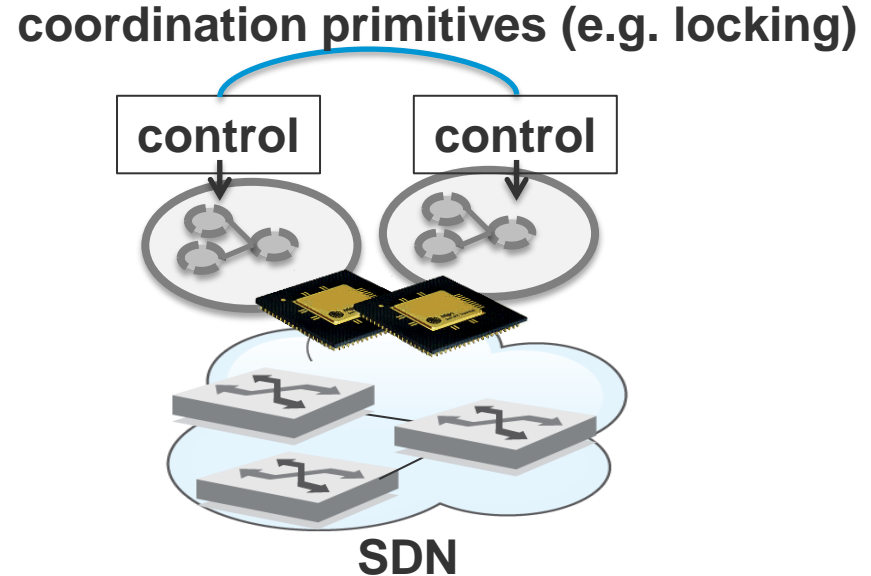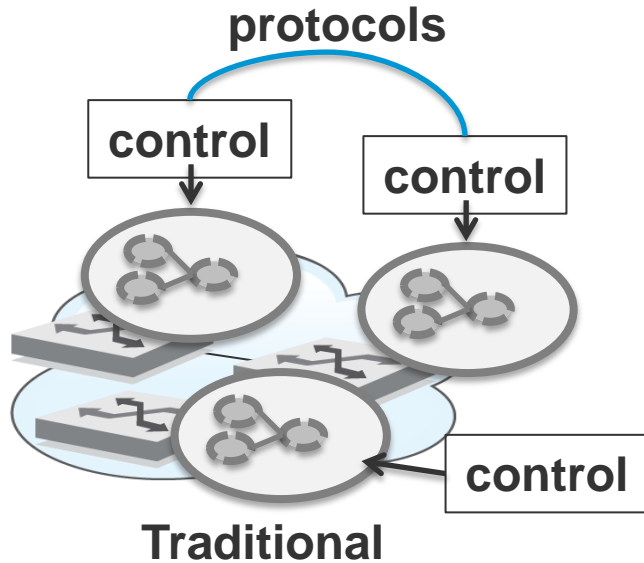
- Programming models very similar in complexity
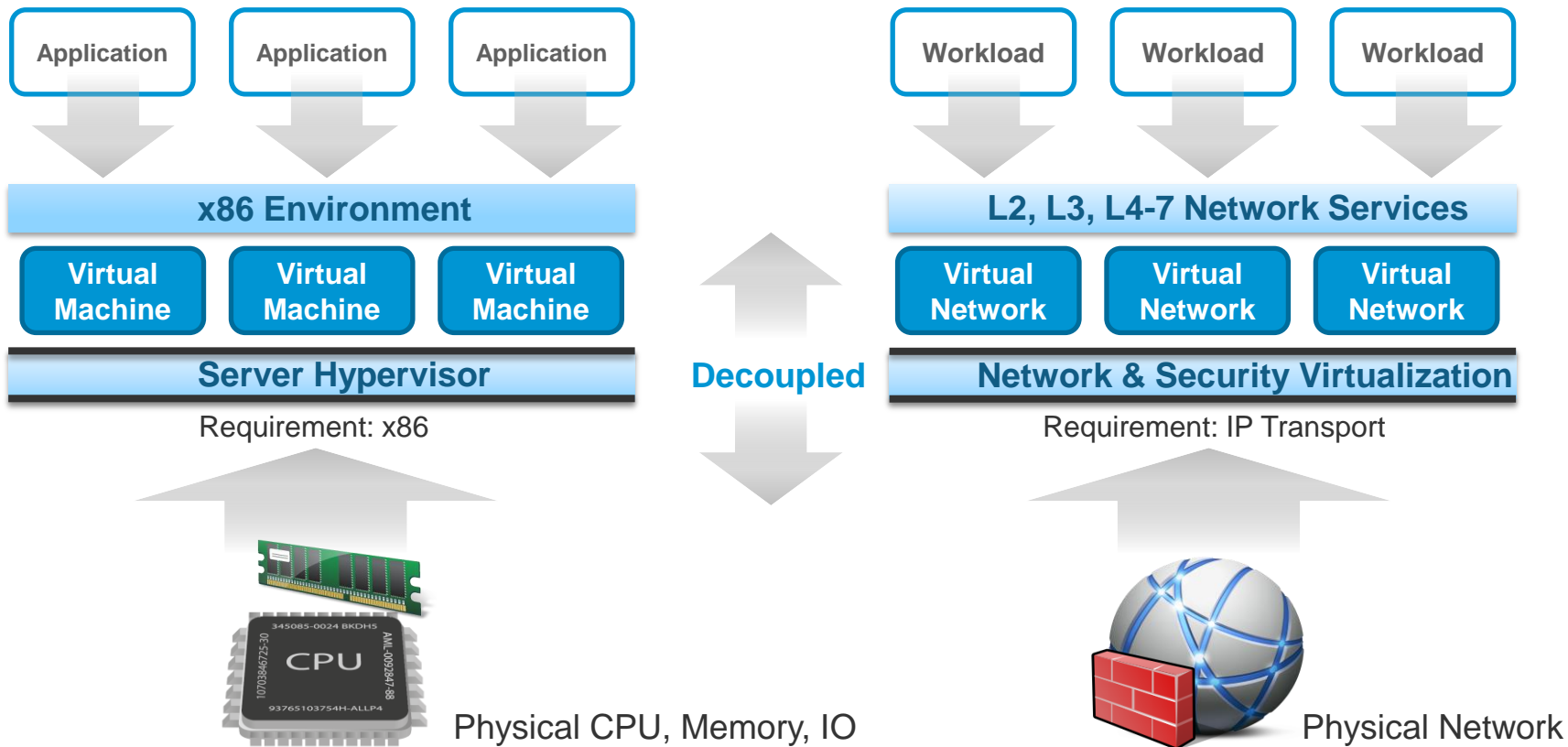- SDN provides **more control, better guarantees, not necessarily simplicity**

Does SDN simplify operations?

- **Separation of control plane from data plane does not inherently lead to operational simplification**

  - You still have the same number of devices to manage, the same amount of network state

- **Network Virtualization, however, is all about operational simplification…**
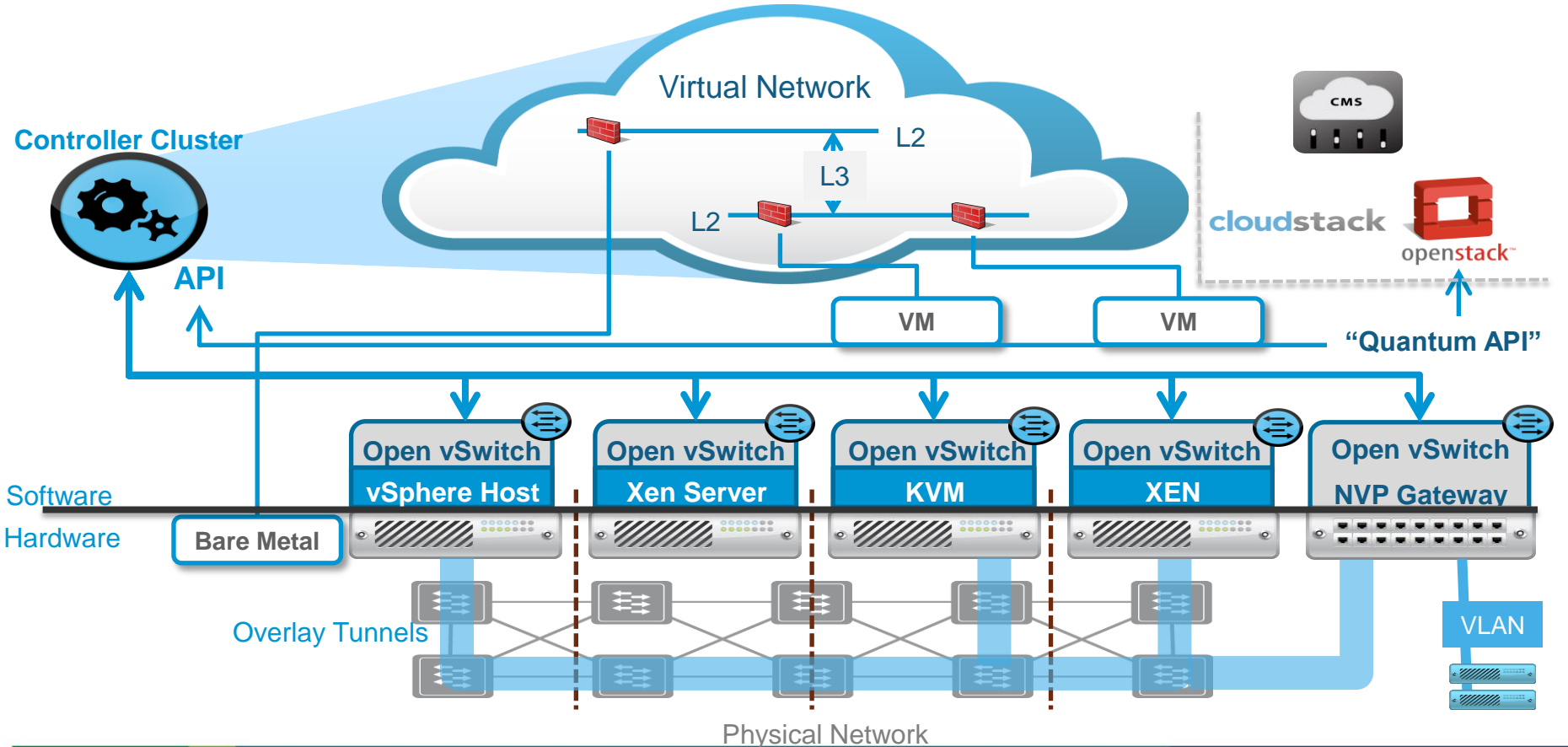
vmware®

What Network Virtualization is

- **Decoupling of the services provided by a (virtualized) network from the physical network**

- **Virtual network is a "container" of network services (L2-L7) provisioned by software**

- **Faithful reproduction of services provided by physical network**
 - Analogy to a VM – complete reproduction of physical machine (CPU, memory, I/O, etc.)

vmware®

Network Virtualization

| Application | Application | Application | | Workload | Workload | Workload |

x86 Environment

| Virtual Machine | Virtual Machine | Virtual Machine |

Server Hypervisor

Requirement: x86

L2, L3, L4-7 Network Services

| Virtual Network | Virtual Network | Virtual Network |

Network & Security Virtualization

Requirement: IP Transport

Decoupled

Physical CPU, Memory, IO

Physical Network

A Network Virtualization Platform

SDN != NETWORK VIRTUALIZATION
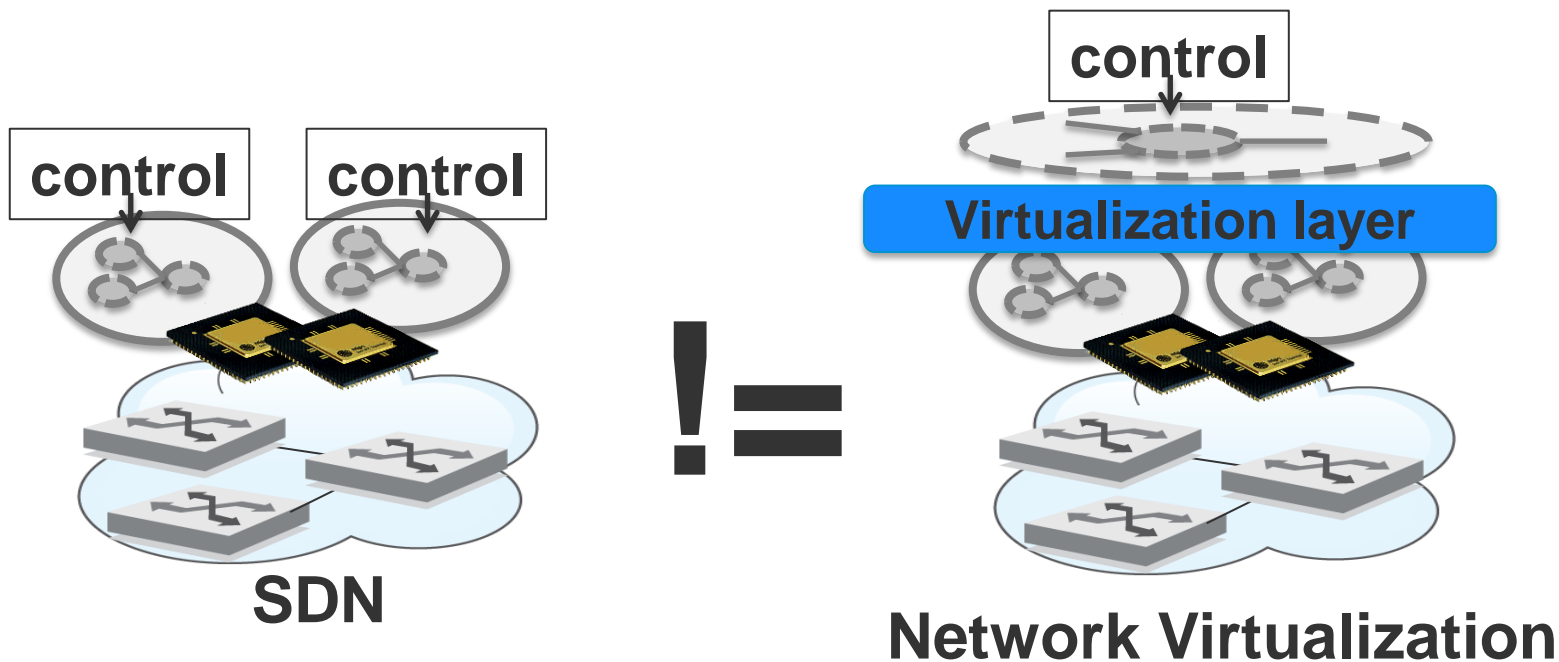
SDN alone doesn't abstract away details of physical network

SDN not required for network virtualization, but useful

About those benefits of ~~SDN~~ Network Virtualization

- **Rapid innovation: network services now delivered at software speeds (vswitch + controller)**

- **New forms of network control: API to NV controller allows creation and management of virtual networks under software control**
  - Snapshot, rollback etc.

- **Vendor choice: decoupled networking services from underlying hardware**

- **Simplified programming: expose abstractions that make sense to programmers**

- **Simplified operations: network state managed like a VM – provision centrally in SW, snapshot, rollback, etc.**

vmware®

Operations and Management

- Maintain existing interfaces
  - SNMP
  - Port counters
  - Port mirroring
  - sFlow
- Plus the power of virtualization
  - Programmatic Provisioning
  - Full configuration snapshot/rewind
  - Network-wide upgrade
  - Accurate accounting and billing
  - Manage at the granularity of virtual networks

vmware®

Conclusions

- **SDN: a mechanism, not a panacea**

- **SDN != Network virtualization**

- **Network virtualization may use SDN, not strictly required**

- **Network virtualization delivers:**

  - Decoupling of network services from hardware

  - Vendor choice

  - Software innovation speed

  - Application deployment times shrink from days to minutes

  - Virtual networks managed programmatically, just like VMs

vmware®