Software Defined Networking
Subtitle:
Network Virtualization

Terry Slattery
Chesapeake NetCraftsmen
Principal Consultant
CCIE #1026

What is Virtualization?

- **Virtual**

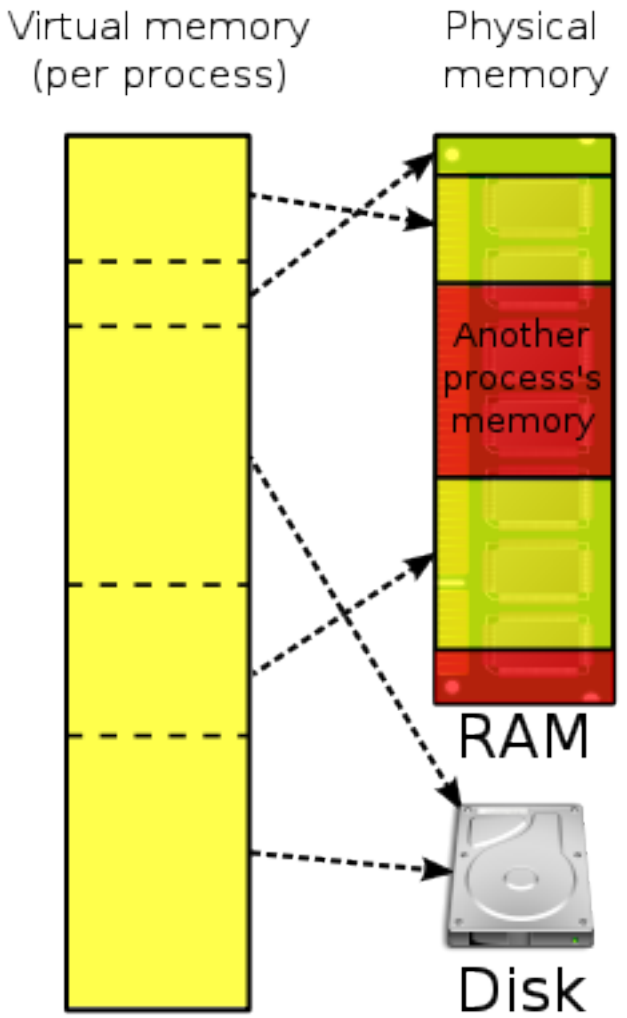
  Existing or resulting in essence or effect though not in actual fact, form, or name: *the virtual extinction of the buffalo. (The Free Dictionary)*

  Not physically existing as such but made by software to appear to do so: "virtual images". (Google)

- **Something that you can use as if it were real.**
  - **Virtual memory and virtual disks are used as if they are real, but are built from multiple underlying components that may be different than the physical entities.**
  - **Driven by queuing theory: shared resource pools are more efficiently utilized than individual pools**

Memory Virtualization

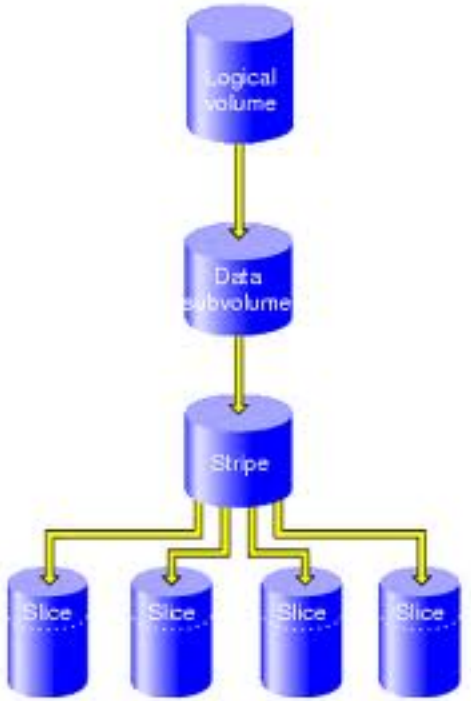
- **Precursor was memory overlays**
  - **Programmer designed overlays and controlled transitions**
- **VM made programming more efficient**
  - **Automatically handled loading data and instructions into RAM**
  - **LRU algorithms balanced inefficiencies of manual tuning**
  - **Programmer efficiency increased**
  - **Abstraction on top of physical memory**
- **Hides complexity**



Wikipedia Image

Disk and Storage Virtualization

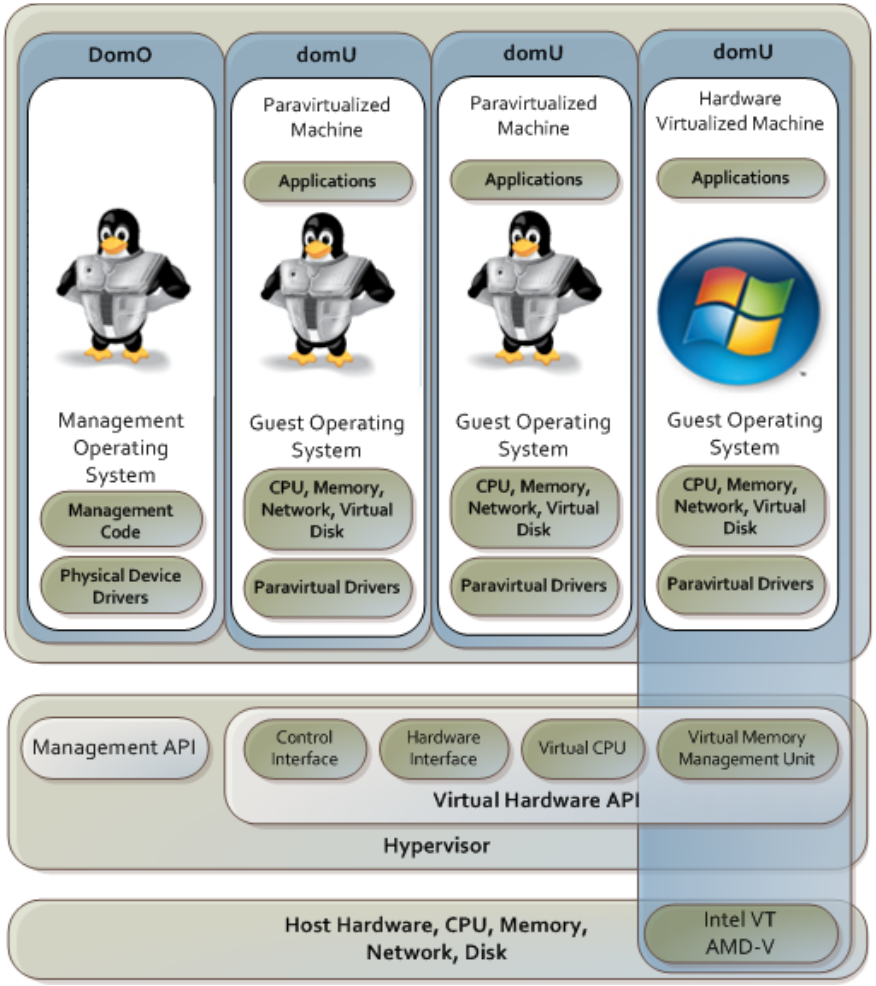
- In the old days…
  - Similar to old memory allocation mechanisms
  - Admins specified disk partitions & assigned data
  - Required advanced planning and usage estimates
  - Changing partition size was a manual process
- Virtual disk partitions
  - Resize by adding/removing slices
  - Increase of administrator efficiency offsets mapping to physical disk
  - Storage abstraction on top of physical disk space
- Hides complexity



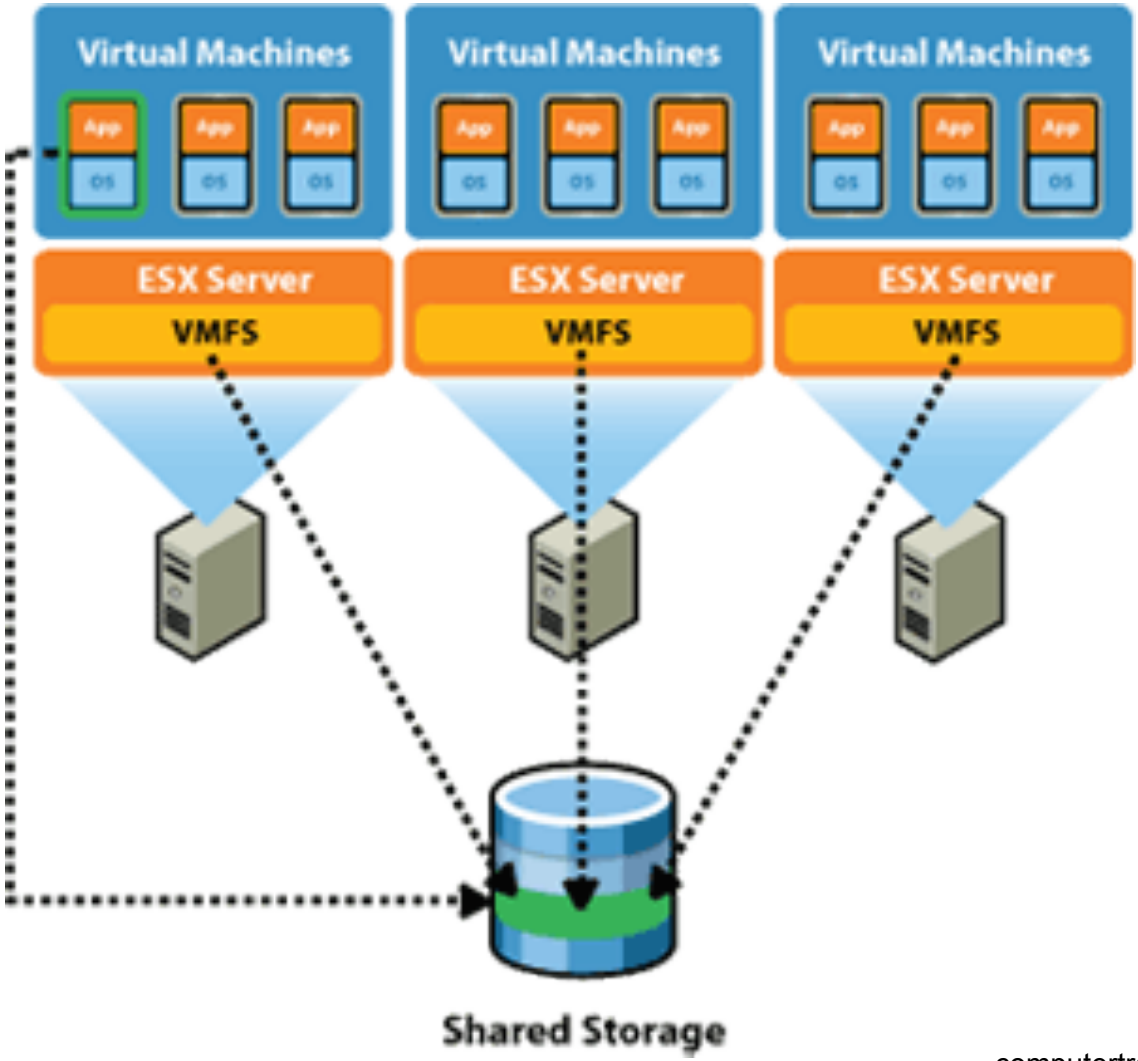
CPU Virtualization

- **Large resource pool drives greater usage efficiency**
- **Abstraction of Virtual Machines running on physical compute clusters**
- **Use of "stock" OS installations improves system administrator efficiencies**
- **Hides complexity**

oracle.com



Combining Virtualization



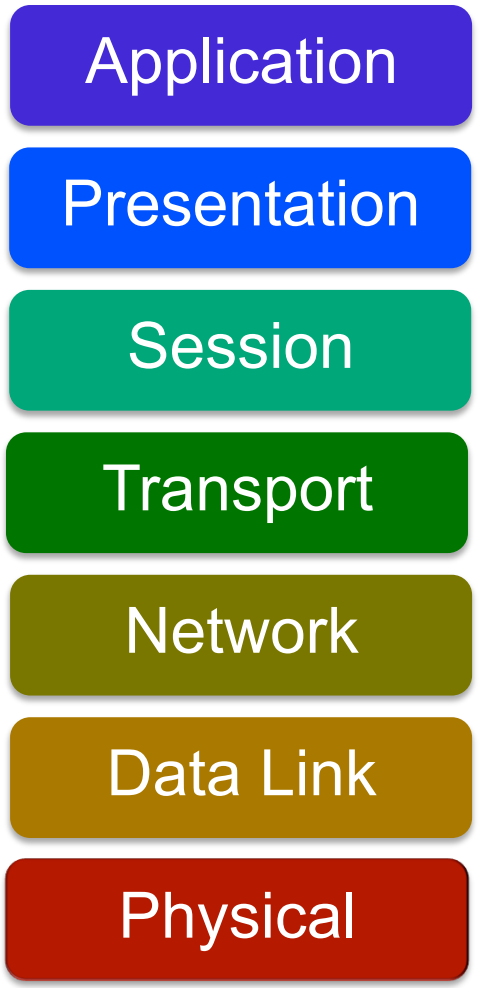
computertraining2011.blogspot.com

What About Network Virtualization?

- L2 (Ethernet example)
  - Start with coax
  - Multi-port transceivers (Cabletron)
  - Multi-LAN chassis (Cabletron)
  - VLANs
  - Q in Q, VXLAN, NVGRE, etc
- L3
  - MPLS (and other L3 tunnel technologies)
- L2-L4 abstraction - simplifies networks and hides complexity?

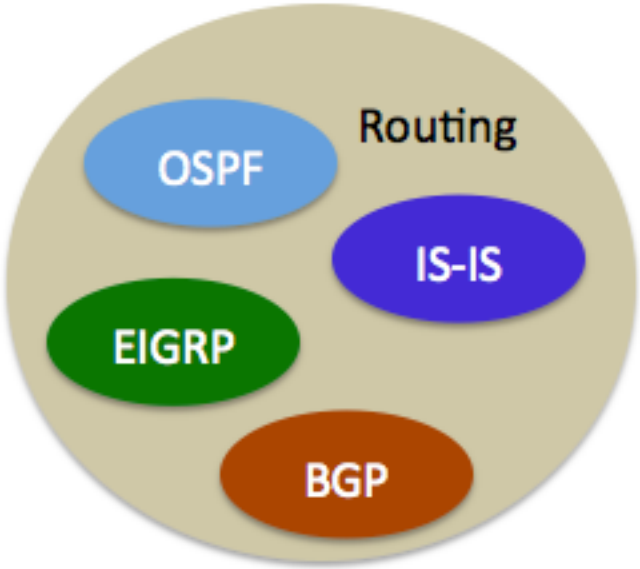
Data Plane Abstractions

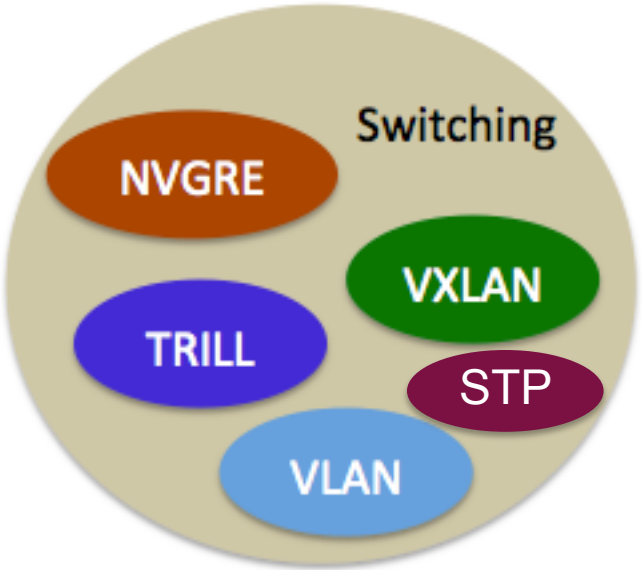
- **OSI data layering model**
- **Some inefficiency**
- **Simplifies design and implementation**
- **Hides details and complexity of lower layers**



Control Plane Abstractions

- No layering
- Complexity is not hidden
- Baroque interfaces between protocols





Combining Network Virtualization

- **Needed: control plane + data plane abstractions**
  - **Create a L3 domain to handle Internet HTTP to N servers in data center 1, with basic security and load balancing**
  - **Add/remove servers to the Internet HTTP domain as load changes**

- **Opportunity: Merge with compute virtualization?**
  - **More powerful and more useful abstractions**
  - **Implies greater ease of use (lower admin effort/cost)**

What Is SDN?

- **Network virtualization**
  - Create control plane abstractions
  - Hide complexity
  - Cleaner interfaces
  - Cost: some network efficiency lost
  - Benefit: Stability, efficiency of use
  - Think: VMware for networking
  - Decoupling the logical from physical resources

The future of networking lies in cleaner abstractions.

SDN is merely a set of abstractions for the control plane.

– Scott Shenker

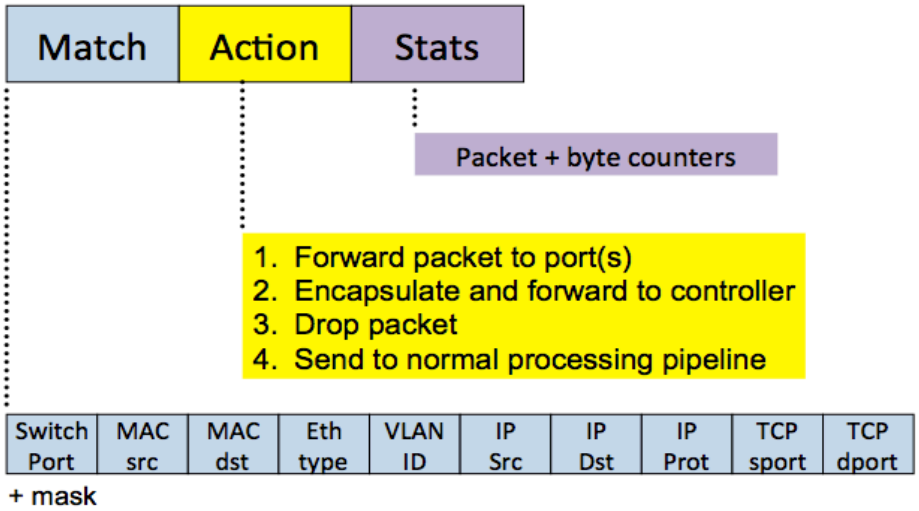
OpenFlow

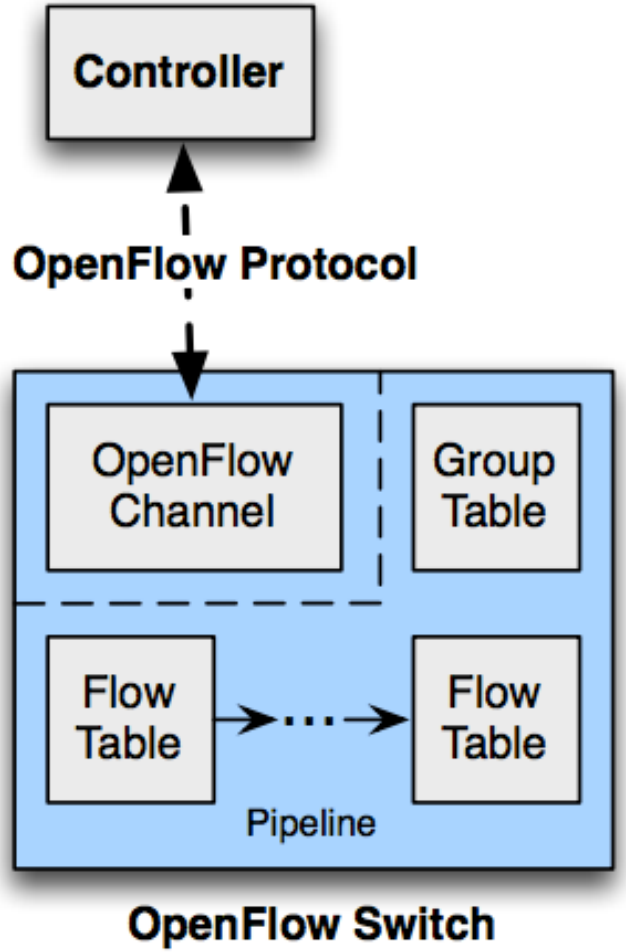
- SDN started with OpenFlow
- API to allow apps to program forwarding tables in switches
- Relatively new protocol
  - ACM paper by Nick McKeown, *OpenFlow: Enabling innovation in campus networks*, April 2008
  - OpenFlow 1.0: Dec 31, 2009
  - OpenFlow 1.3.1: Sept 2012
- Centralized controllers are not new

OpenFlow doesn't let you do anything you couldn't do on a network before." – Scott Shenker

OpenFlow

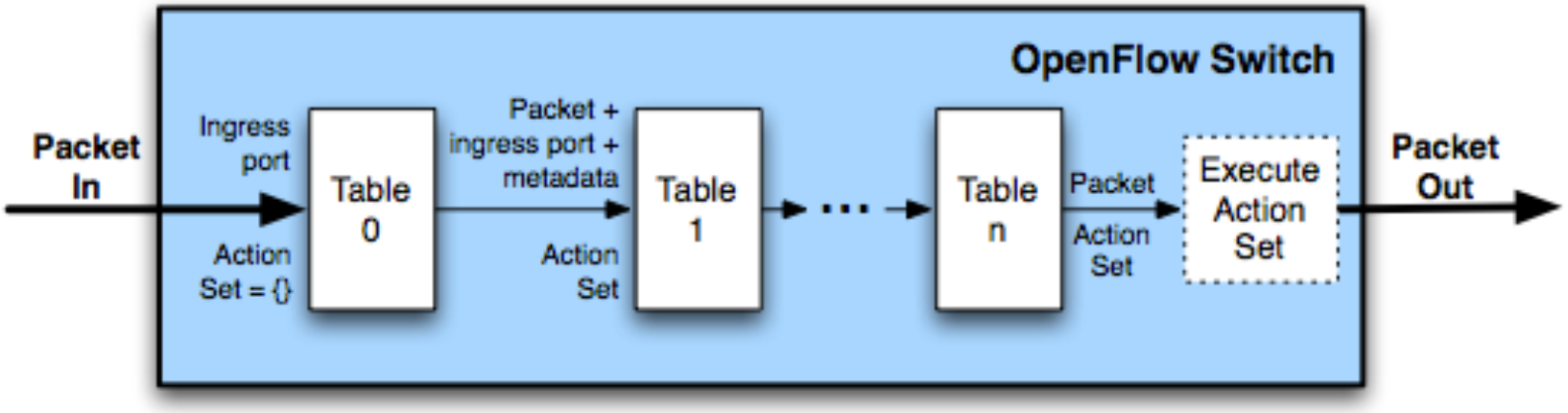
OpenFlow is an open standard that enables researchers to run experimental protocols in the campus networks we use every day.
- openflow.org



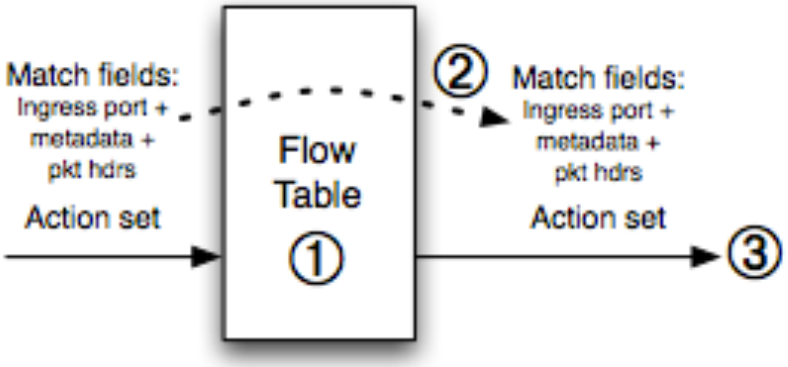


openflow-spec-v1.1.0

OpenFlow Processing



(a) Packets are matched against multiple tables in the pipeline



① Find highest-priority matching flow entry

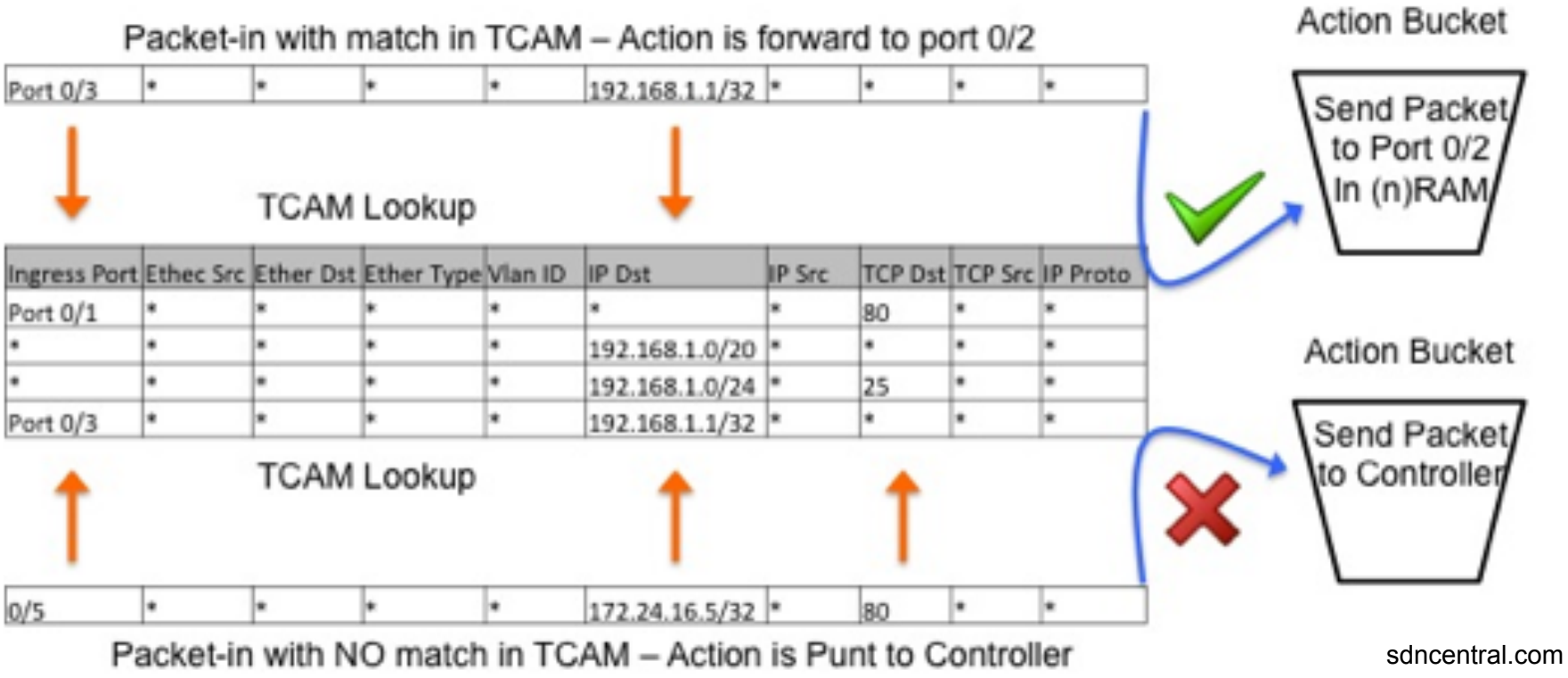
② Apply instructions:
- i. Modify packet & update match fields (apply actions instruction)
- ii. Update action set (clear actions and/or write actions instructions)
- iii. Update metadata

③ Send match data and action set to next table

(b) Per-table packet processing

openflow-spec-v1.1.0

OpenFlow Packet Forwarding Engine



sdncentral.com

- **Match, Action**
  - **Actions: forward, drop, push/pop a new header, modify header fields, or forward to controller**
- **Counters kept on all flow entries**

OpenFlow Designed for Research

Will OpenFlow Scale for Production?

OpenFlow Limitations

- **Insufficient functionality**
  - Need non-flow configuration (see OF-Config 1.1)
  - Need new abstractions to simplify networking and reduce the potential for errors
- **Scaling problems**
  - Per-flow processing in a big DC (10M flows/sec?)
  - Multiple control points (flow rate X control points)
  - TCAM size limits (particularly in ToR switches) (bradhedlund.com & ioshints.info)
  - Scaling mechanisms will need to be developed

What Is SDN?

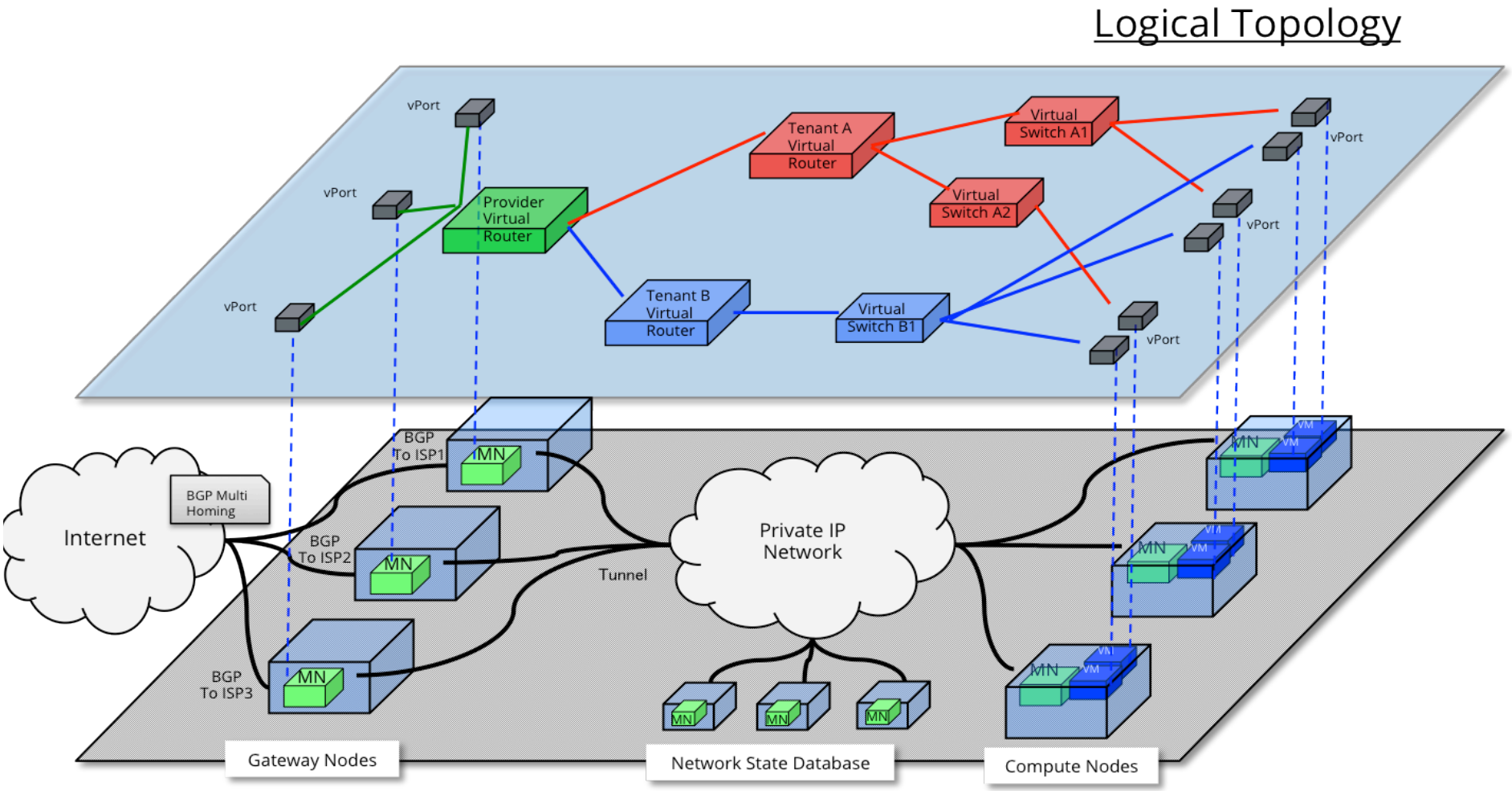
- Means different things to different people
- It is NOT OpenFlow!

- It is a paradigm shift

- My definition…
  - High level abstraction of the control plane
  - Virtualize the network
  - Can work with the network on a conceptual basis without mapping to the physical elements

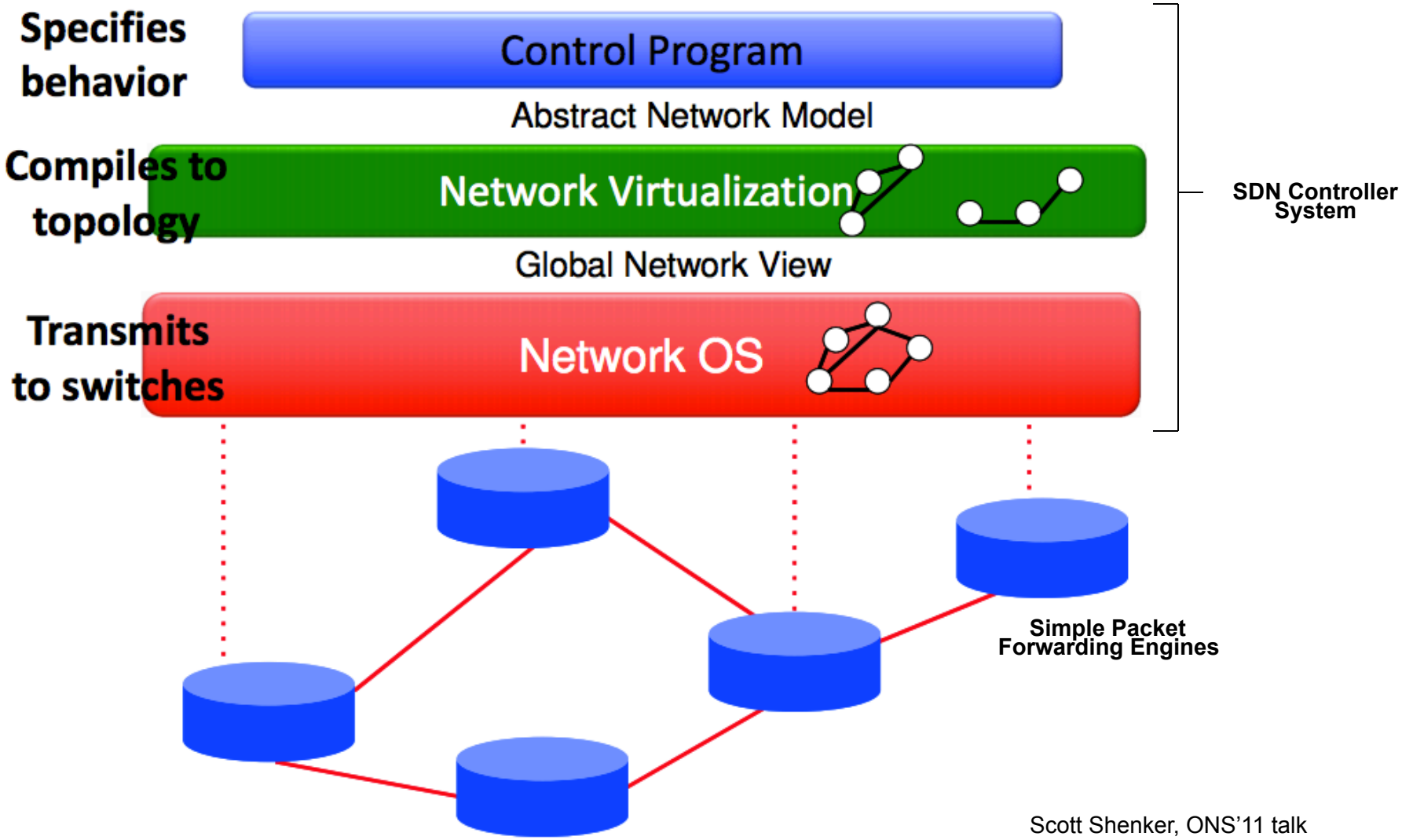
- Implication: It changes the deployment and business models

True Network Virtualization



MidoNet solution diagram provided by Midokura

SDN Anatomy



Scott Shenker, ONS'11 talk

Network Virtualization

- **Migrate L2-L4 along with the VM**
  - **Migrate Load Balancers and Security**
- **It's why VMWare bought Nicira for $1.2B**
- **Example:**
  - **How does traditional SNMP counter handling work with VM migration?**
  - **Need to move counters along with the CPU, Memory, Storage, and Network**

SDN Guidelines

- **Handle complexity (ACL, QoS, mobility) at the edge (in vSwitches)**
- **Overlay the physical network with a virtual network**
- **Switches *may* use tunneling to forward packets**
- **Don't need to upgrade your hardware switches**
- **Controller is** logically **centralized**

We'll see how this turns out…

SDN Hype

- **Centralized management and control of multi-vendor networks**
  - **Redundant controllers - split brain operation**
  - **In-band or dedicated management network?**
  - **What about linecard protocols (e.g., BFD)?**
  - **Scaling issues to be identified and solutions developed**
- **Uniform policy deployment**
  - **Requires uniform policy definitions**
  - **Existing configuration management systems marginally successful; changing the mechanism won't fix it**
  - **UI and API to define policy and exceptions**
  - **Better QoS and TE configurations?**

SDN Hype

- **Fewer configuration errors**
  - Errors propagate faster; bigger impact
  - Controllers must be smarter to avoid common errors
  - Configuration library is needed
    - Similar to software development abstractions
    - Eliminate sources of errors
- **Increase scalability and optimum forwarding**
  - Per-flow forwarding decision making doesn't scale
    - RTT to controller is too expensive
    - Fallback operation if controller doesn't respond
    - Use aggregate flow entries
    - Other optimizations To Be Developed

SDN Hype

- **Integrated security**
  - **It Depends™**
  - **How complete a solution? Pure OpenFlow?**
  - **Basic security is possible**
  - **Virtual appliances with a virtual network overlay are a more complete solution**

- **Load balancing**
  - **Load balancer built in 500 lines of code**
    YouTube: *Aster*x: Load-balancing as a network primitive,* Nikhil Handigol
  - **No additional hardware; just "smart routing"**

SDN Hype

- **Per-tenant QoS**
  - Certainly at the edge
  - Must still handle shared BW on aggregation links
- **Expect vendor extensions**
  - Differentiation between vendors
  - Customer lock-in

Is SDN a Fad?

- East-West flows dominate DC traffic
- Shared resource pool is more efficient
- Rate of change at the edge is increasing
  - But the network's ability to effect change is lagging
- Need automated, multi-vendor methods for network configuration management
  - CLI isn't sufficient

What I See Coming…

- SDN is not a fad
  - It will be different than the current hype
- Good benefits
- Worth the pain of transition
  - The current pain makes it worth the transition
- Hides network complexity (doesn't reduce it)
- Don't throw out good network design practices
- Managing an SDN will be different

System View of the Network

- **We've needed a system view of the network**
  - **Difficult with device-centric systems**
- **Logically-centralized system**
  - **Central point of control**
  - **Should be physically distributed**
- **Examples:**
  - **Network-wide QoS with a consistent UI**
  - **Load balancing when and where you need it**

Improved Traffic Engineering

- **Central view of traffic engineering**
  - **Direct traffic where you want it and via which links**
  - **Routing protocols "pull" traffic**
  - **Policy routing is too manual and device-centric**
  - **Google improved WAN utilization (40% to 90%+)**
- **Load distribution over many paths**
  - **Central controller can use historical flow information**

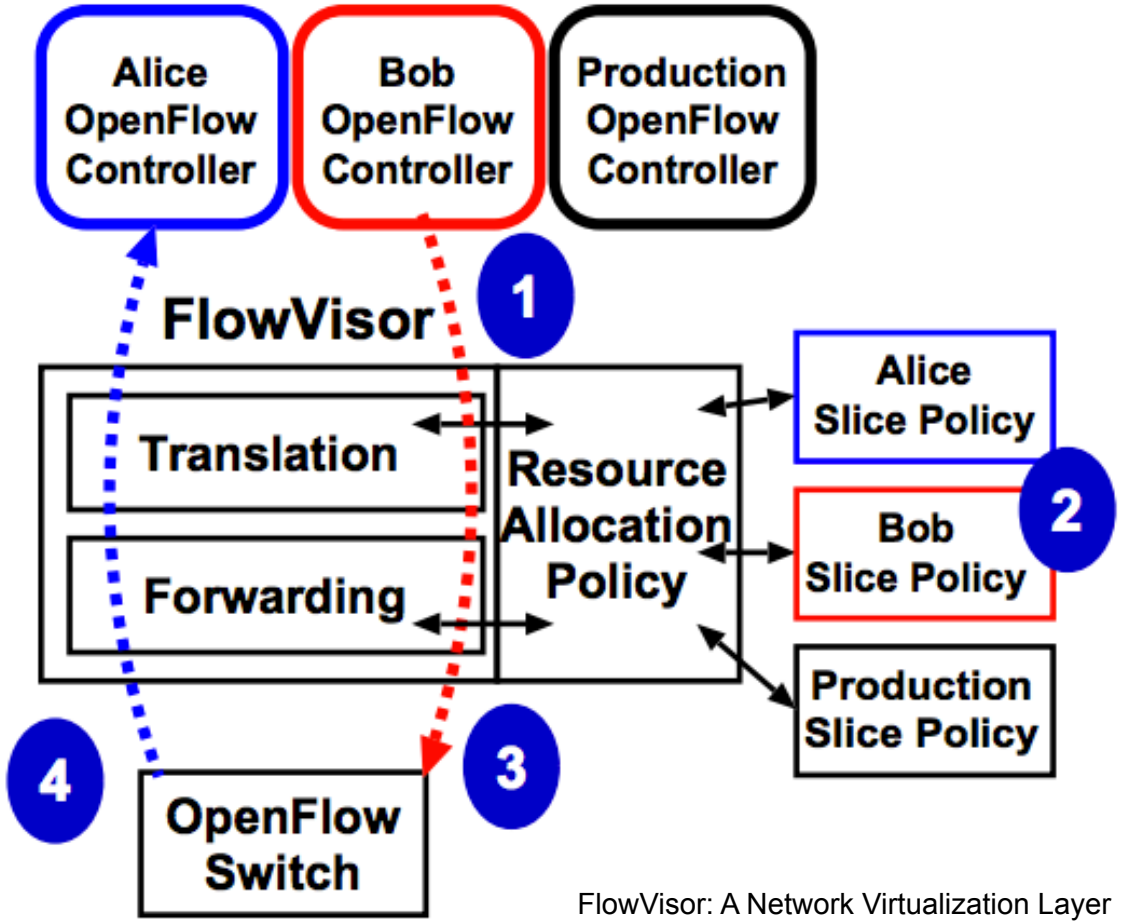
New Protocol Development

1. Intercept OpenFlow messages (both directions)

2. Policy check
   - Which slice?
   - Valid operation?

3. Forward message (rewrite if needed)

4. Pass return messages to the correct controller



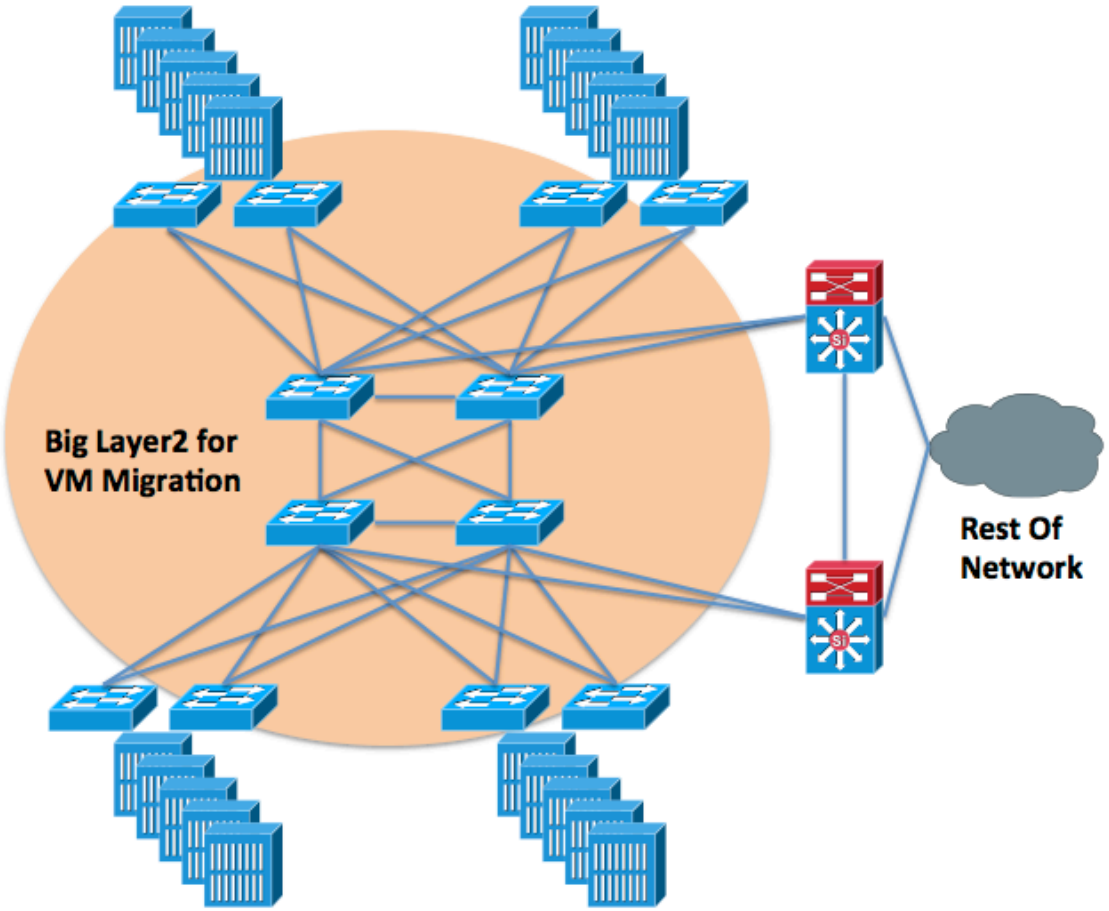
FlowVisor: A Network Virtualization Layer

Rapid Provisioning and Migration

- What's your provisioning time? Migration time?
- Building an agile data center
- UI + API to provision CPU, Memory, Storage, & Network together
- Easily move workloads for energy savings
  - Turn off unused switch ports as well as servers

Use Cases – Reduce Data Center L2

- Reduces the need for large DC L2 domains
- Overlay L2-L4 over a well-designed network



Use Cases – Multi-tenant Networks

- Use their own L3 addressing
- Virtual overlay networks
- Dynamic remapping of the ports in the virtual networks
- Avoids VLAN number exhaustion
- Alternatives can be made to work
  - VXLAN, NVGRE
  - Aren't as comprehensive

Tenant 1
Tenant 2
Tenant 3

Other Use Cases

- **Energy reduction practices**
  - **opennetworking.org video**
- **IPv6 address tracking**
  - **ioshints.info tip**
- **SPAN traffic selection and director**
- **Provisioning for Big Data analysis**
- **Observation: multiple technologies to address the suite of use cases**

Integration With the Rest of the Network

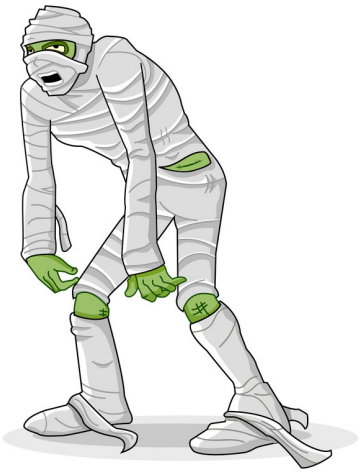
- **Run L2 or L3 protocols at the edge**
  - **SDN cloud can look like one router/switch**
  - **Controller runs routing protocols**
  - **Switches forward routing protocol packets to controller**
  - **Expect "*interesting*" failure modes and bugs**

Troubleshooting Will Change

- Controller connectivity problems
  - In-band path to switches
    - Connectivity may need to be repaired first
    - <diagram of problem?>
  - Out of band path to switches
    - Separate network to provision and manage
    - Use SDN with in-band communications on control network? (Vicious cycle?)
  - Split brain situations

Do We Need SDN?

- **Network Configuration**
  - Manual processes don't scale
  - Long deployment times
  - Inconsistent policy implementation
  - Multi-vendor, typically via CLI, is *hard*
- **Multiple technologies to achieve similar solutions**
  - Interactions between and support of the technologies
  - Layering functions on top of one another; additional complexity



Will SDN Eliminate Jobs?

- **Not likely**
- **Daily workload will change**
  - **Software and scripting experience will help**
- **Shift to more valuable tasks**

Proof of Concept

- **Begin experimenting with SDN**
  - Begin learning some of the lessons
  - Evaluate controllers and switches
  - Improve corporate IT systems agility
  - $100K - $500K cost (switches, controller, staff cost)
  - Ongoing platform for evaluating and debugging network-aware apps
- **Begin organizational transition**
  - Culture
  - Developing lines of communication
  - Proactive adoption

Predictions

- Some SDN protocols will run in the network device
  - Local decisions for non-stop operation, performance, and scaling
  - Line card protocols will run locally, communicating with SDN controller
- Scaling issues will be addressed
  - May operate like IPmc – flow starts on the default path, then switches to the optimum path.
- Virtual networks will simplify the common case
- New failure modes and troubleshooting tips
- Widespread adoption in 5 years

Questions I Have

- What do the controller abstractions look like?
  - Defining a group of devices/interfaces to apply a policy
  - What do policies look like?
    - Set all interfaces with characteristic X to 100/Full
    - Map flows to web server Y to distributed cluster Z
    - Apply QoS/security policy to all interfaces like X
- System monitoring and management
  - SNMP isn't sufficient - slow to develop/change
  - Do the abstractions match?
  - How are error conditions reported?

Summary

- SDN is a disruptive technology
- It will look different than it does today
- New design rules, new challenges
- Big changes ahead
- It will be an exciting journey

Questions?

- **Further Reading…**
  - **http://www.nec-labs.com/~lume/sdn-reading-list.html**
  - **Network Virtualization**
    - **bradhedlund.com**
    - **blog.ioshints.info**
    - **Scott Shenker – Gentle Introduction to SDN (YouTube)**
  - **OpenFlow**
    - **opennetworking.org**
  - **Georgia Tech SDN MOOC – coursera.com**
  - **"Enterprise Data Center Security with Software Defined Networking" – IBM pdf**
  - **http://www.imsaa.org/tutorial_4.pdf**

Terry Slattery
Chesapeake Netcraftsmen