Network Abstraction
The Network Hypervisor

David Hadas,
Haifa Research Lab,
Nov, 2010

Davidh@il.ibm.com

Agenda

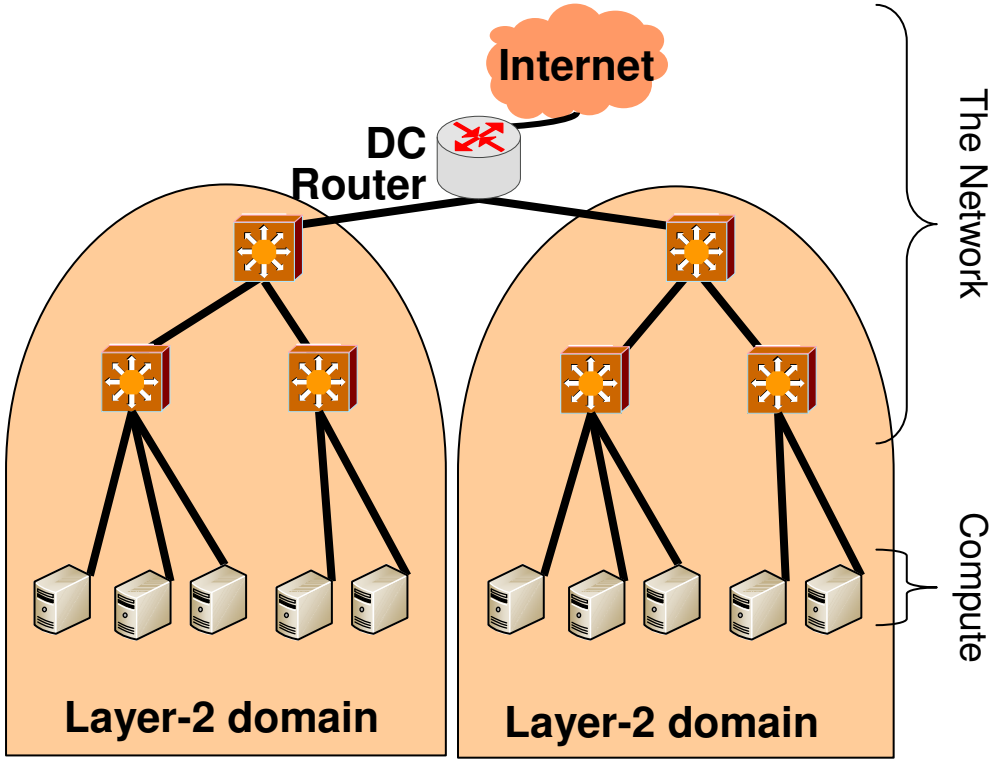
- New Requirements from DCNs
  - Virtualization
  - Clouds
- Our Approach: Building Abstracted Networks
- Virtual Application Networks[1,2] (VANs)
  - An Abstracted Network solution
    Developed by HRL as part of Reservoir
  - Abstracted Network Challenges

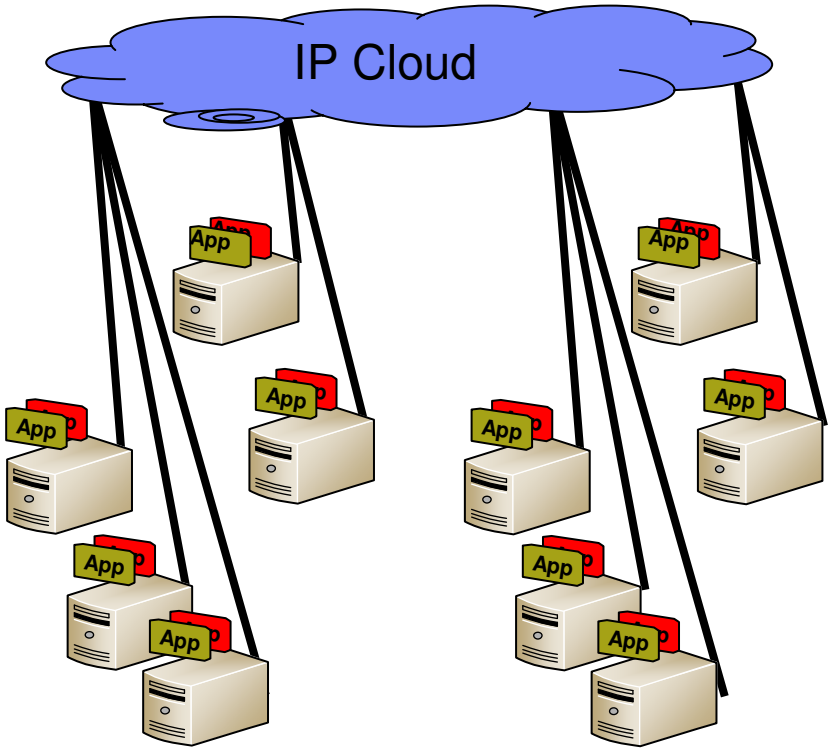
(1) D. Hadas, S. Guenender, and B. Rochwerger, "Virtual network services for federated cloud computing," tech. rep., IBM, 2009. Research Report H-0269.

(2) A. Landau, D. Hadas, and M. Ben-Yehuda, "Plugging the hypervisor abstraction leaks caused by virtual networking," in SYSTOR '10: Proceedings of the 3rd Annual Haifa Experimental Systems Conference, pp. 1–9, ACM, 2010.

Traditional Network View

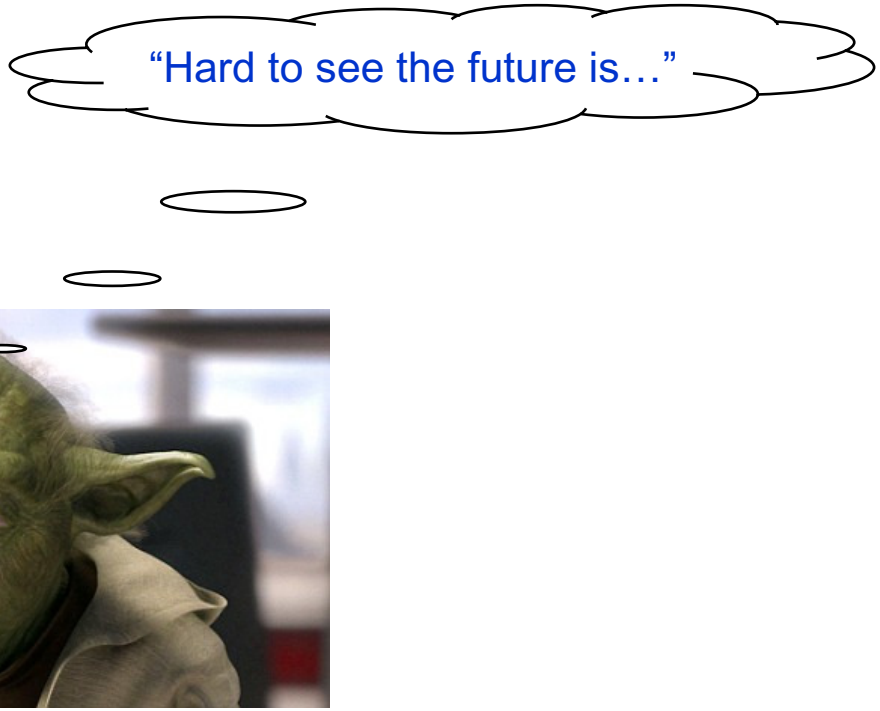
- Network View
  - DCs scale to 100,000 End Stations[1,2]
  - Routers connect L2 domain, sub-divided to VLANs
  - Typically a LAN is up to 250 End Stations[1,2,4]
  - Typically a L2 domain is up to 4K End Stations[1,2,3]
  - Scale: 1000 LANs, 100 L2 Domain
- Server View
  - End-Stations have single IP and MAC and are "dumb" (know only a default GW)
  - Network topology and routes are under the control of the network





[1] Albert Greenberg , Parantap Lahiri , David A. Maltz , Parveen Patel , Sudipta Sengupta, Towards a next generation data center architecture: scalability and commoditization, Proceedings of the ACM workshop on Programmable routers for extensible services of tomorrow, August 22-22, 2008, Seattle, WA, USA
[2] Greenberg, A., et al., "The Cost of a Cloud: Research Problems in Data Center Networks", CCR, v39, n1, Jan'09.
[3] C. Kim and J. Rexford. Revisiting ethernet: Plug-and-play made scalable and efficient. In Proc. IEEE LANMAN Workshop, 2007.
[4] P. Garimella, Y.-W. E. Sung, N. Zhang, and S. Rao. Characterizing vlan usage in an operational network. In INM '07

The Future of DCNs



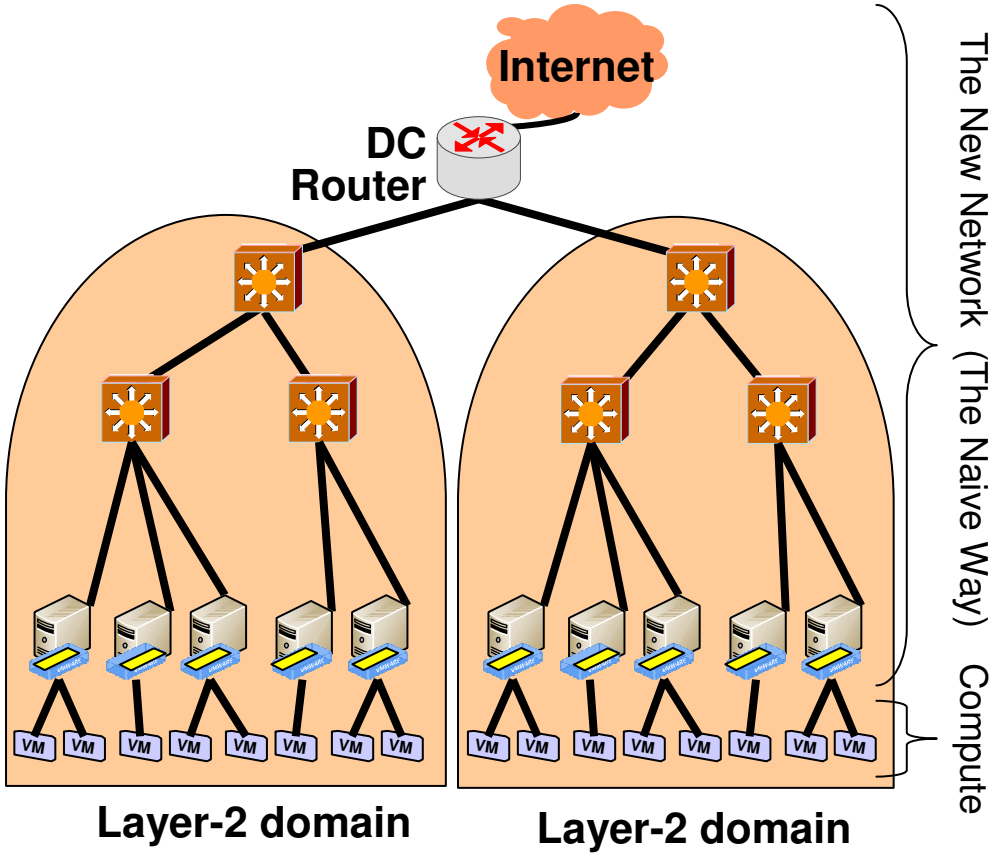
- # of Physical Hosts per DC is already over 100,000 Hosts
- # of VMs per Host will grow to 64 – 192 (32 core machine)
- 1 or 2 virtual network interfaces per VM
  - We believe the real number is 4, 8 or more since the cost of virtual interfaces is lower than that of physical interfaces, leading to new use patterns
- Total: 10,000,000 – 100,000,000 virtual network interfaces in a large DC!

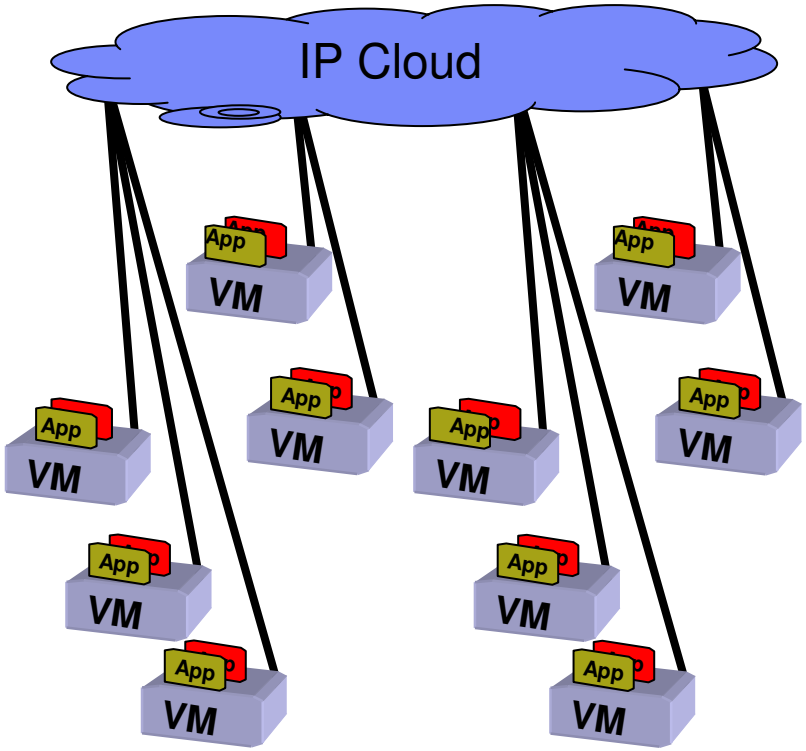
Network Virtualization – The Naive Way

IBM

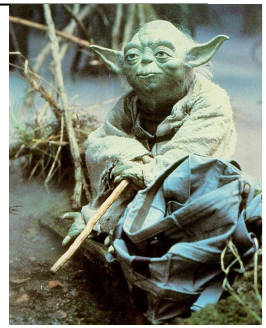
- New Network View
  - The vSwitch is part of the Physical Network!
  - 64-192 VMs per Server (32 Core Machine)
    - Multiple virtual interfaces per VM
  - Ratio: 1:1 LANs to Hosts,
    10:1 L2 Domains to Hosts
  - Scale: 100,000 LANs, 10,000 L2 Domain

- New Server View
  - VMs have single IP and MAC and are "dumb" (knows only a default GW)
  - Network topology and routes are under the control of the network (now spanning into Hosts)





Wouldn't L2 Domains of The Future be Larger?



- Vendors have for many years been seeking ways to grow L2 further with very partial success
  - L2 Domain scaling is on the "most wanted" list for many years
- Ethernet related research (See for example [1,2]) concludes that scaling Ethernet requires fundamental changes into the Ethernet protocol (e.g. to eliminate broadcast services such as ARP)
  - Standardization work has started on IETF and IEEE to address Ethernet related issues (TRILL, Layer 2 Multipathing)
  - Will L2 scale by a factor of 10, 100, 1,000, 10,000?
  - Will L2 fit more complex environments with multiple sites?

[1] A. Myers, T.S.E. Ng, and H. Zhang, "Rethinking the Service Model: Scaling Ethernet to a Million Nodes", in Proceedings of ACM SIGCOMM HotNets, 2004.

[2] Changhoon Kim , Matthew Caesar , Jennifer Rexford, Floodless in seattle: a scalable ethernet architecture for large enterprises, Proceedings of the ACM SIGCOMM 2008 conference on Data communication, August 17-22, 2008, Seattle, WA, USA

Network Infrastructure for Clouds

IBM

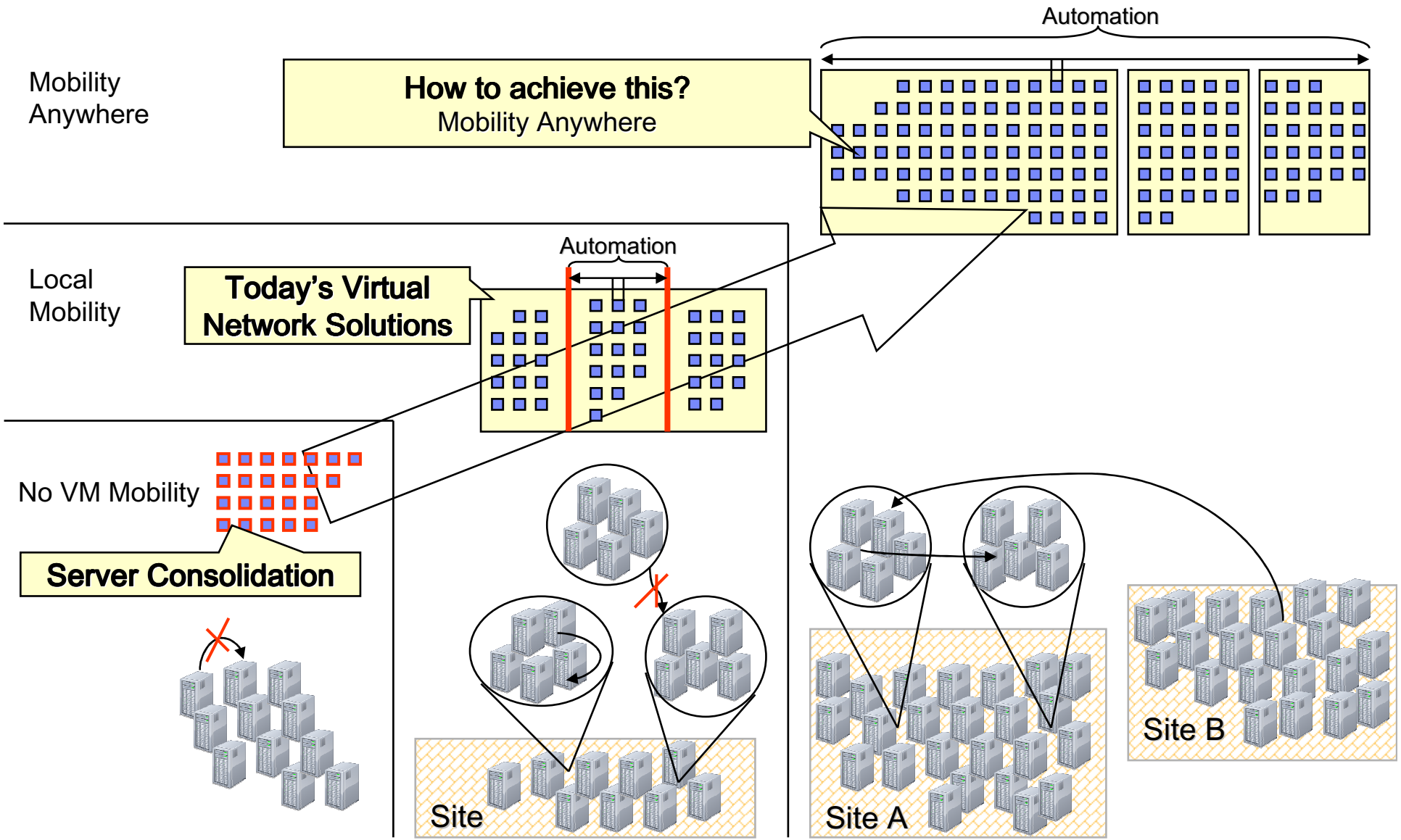
- **Clouds need to scale**
  - # of VMs,
  - # of Vitual networks,
  - # of Sites

- **Clouds need to be agile**
  - Fully automated
  - Enable auto-placement decisions
  - Dynamic
  - Self-managed
  - Simple to use
  - Efficient

Automated Mobility Anywhere

- **Clouds need to support establishing isolated virtual networks**
  - Private to the hosted customer and managed by it
    (allowing customers to determine their own address space)
  - Extend beyond the limits of a single data center (physical and administrative) without requiring coordinated management and while maintaining data center independence and security.
  - Established ad-hoc without requiring pre-configuration and/or management control of the physical network

Mobility Anywhere



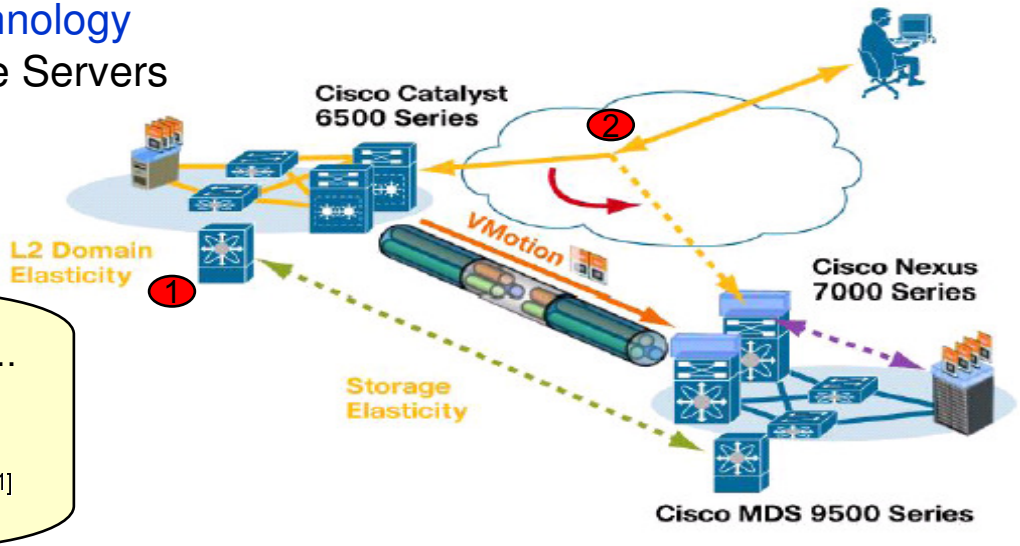
Data Center Virtualization Requires Remodeling Of The Network

IDC, Oct 2009: 'The DC network industry is witnessing a dramatic shift in DC architecture. It is clear that the current network topology will not meet the requirements of the new DC. Server, storage, and network virtualization will be the overriding premise of the DC buildout.'

- A solution that enables customers to freely mobile VMs across Data Centers is needed
  - 1 – A **'Long Distance Virtual LAN'** technology
    - Current approach: Extending local VLANs across subnets
  - 2 – A **Revised Client/Server Routing** technology
    - Connecting Internet Clients to mobile Servers

"(This) is not possible with existing technologies….

(It) will require the redesign of the IP network between the data centers involving the Internet." [1]



[1] http://www.cisco.com/en/US/solutions/collateral/ns340/ns517/ns224/ns836/white_paper_c11-557822.pdf

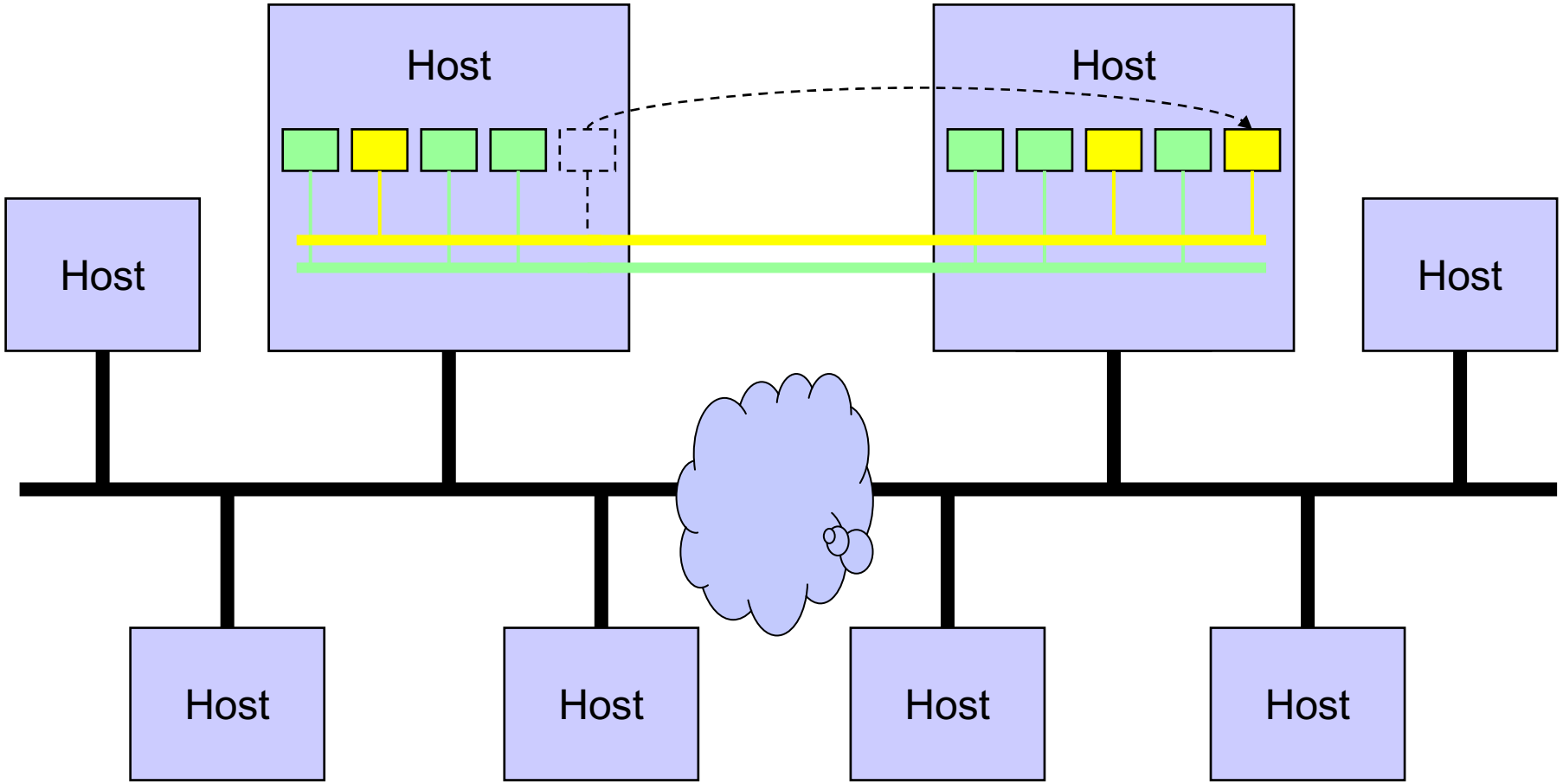
Abstracted Networks – The Network Hypervisor

IBM

- Network virtualization should follow (well established) host virtualization principles:
  - Host virtualization should enable virtual machines
    - To remain independent of physical location
    - To remain independent of the host physical characteristics such as CPU, Memory, I/O, etc.
    - To form isolated compute environments on top of the shared physical host environment
  - Network virtualization should enable virtual machines
    - To remain independent of physical location
    - To remain independent of the network physical characteristics such as topology, protocols, addresses, etc.
    - To form isolated network environments on top of the shared physical network environment serving the hosts

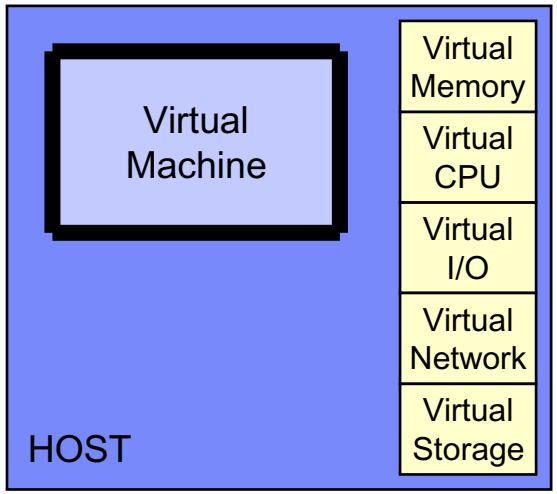
Such complete network virtualization can be achieved using network abstraction – hence the term "Abstracted Networks"

Physical Network Abstraction



Hypervisor Evolution

- In the 1970's, IBM shipped the first hypervisors offering a complete virtual replica of the hosting system
  - Guest virtual machines are offered a fully protected and isolated copy of the underlying physical host hardware
  - Virtual replica can be hardware dependent (No abstraction)!

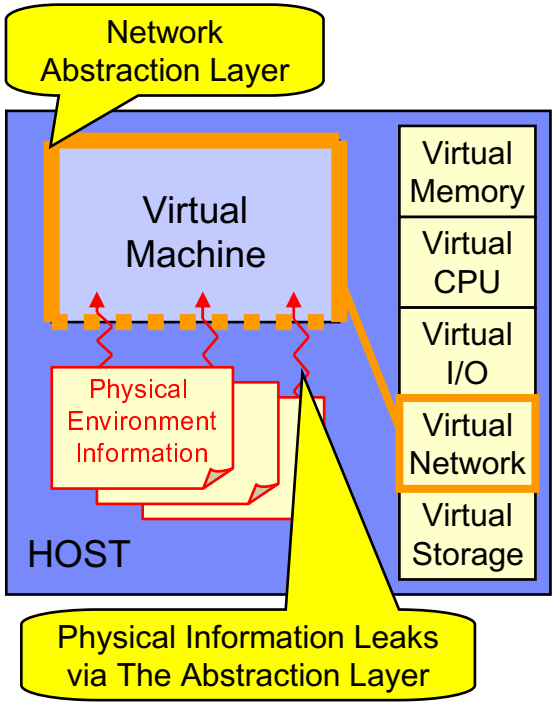


- Later, the hypervisor role evolved in order to support advanced administrative functions, such as
  - Checkpoint/Restart
  - VM Cloning
  - Live-Migration (Mobility)

- The hypervisor needs to ensure that the guest application and operating system are able to continue running unchanged even when the compute, network and storage environments drastically changes.
  - Can be achieved by completely abstracting the environment offered to guests and ensuring that guests become unaware of any characteristic of the hosting platform.

Current Network Abstraction Layers are Leaking

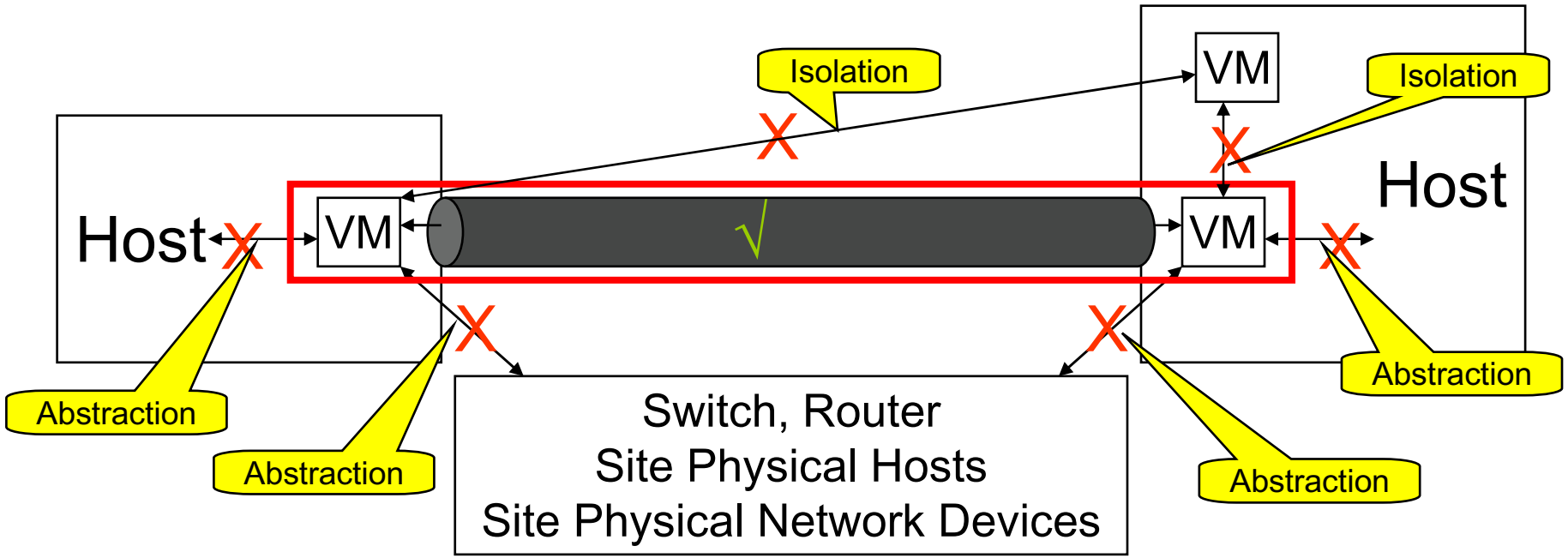
- **Most current hypervisors offer incomplete network abstraction. Common approaches include:**
  - Guest directly accessed a physical network adapter
    - The hypervisor exposes direct references to unique platform hardware resources
  - An emulated NIC or backend is connected to a local host software bridge/router
    - The hypervisor exposes direct references to network addresses with spatial meaning (may also be time-bounded).
  - Use Network Address Translation
    - Received packets continue to include references of physical addresses, causing a leak in the hypervisor abstraction layer.
- Current hypervisors tend to consider this incomplete abstraction as a network problem rather then a problem of the hypervisor.
  - Solution: Restrict the mobility to the network segment
  - Solution: Synchronize the mobility with a timely reorganization of the network.



Plugging The Hypervisor Network Abstraction Layer Leaks

- Abstraction: VMs should not 'talk' to the hardware (physical network)
  - Use separate addresses
  - Hypervisor should provide mapping

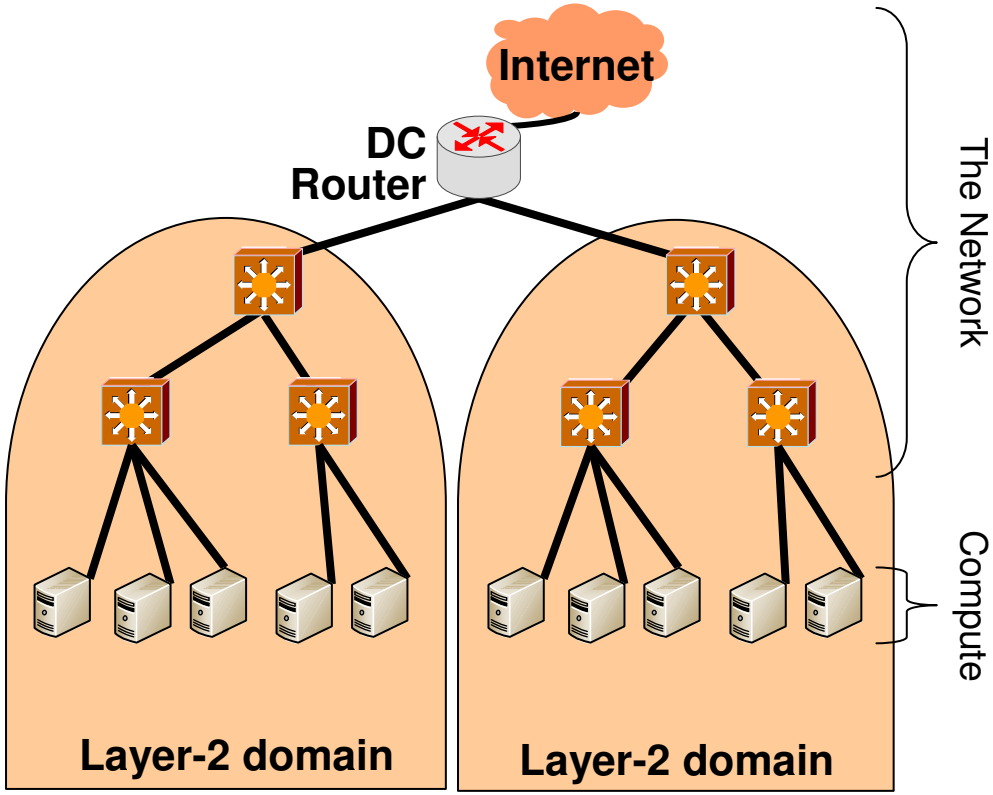
- Isolation: VMs belonging to one virtual network should be isolated from VMs not connected to that network
  - Security by Isolation
  - Isolated Performance
  - Independent Address/Naming schemes

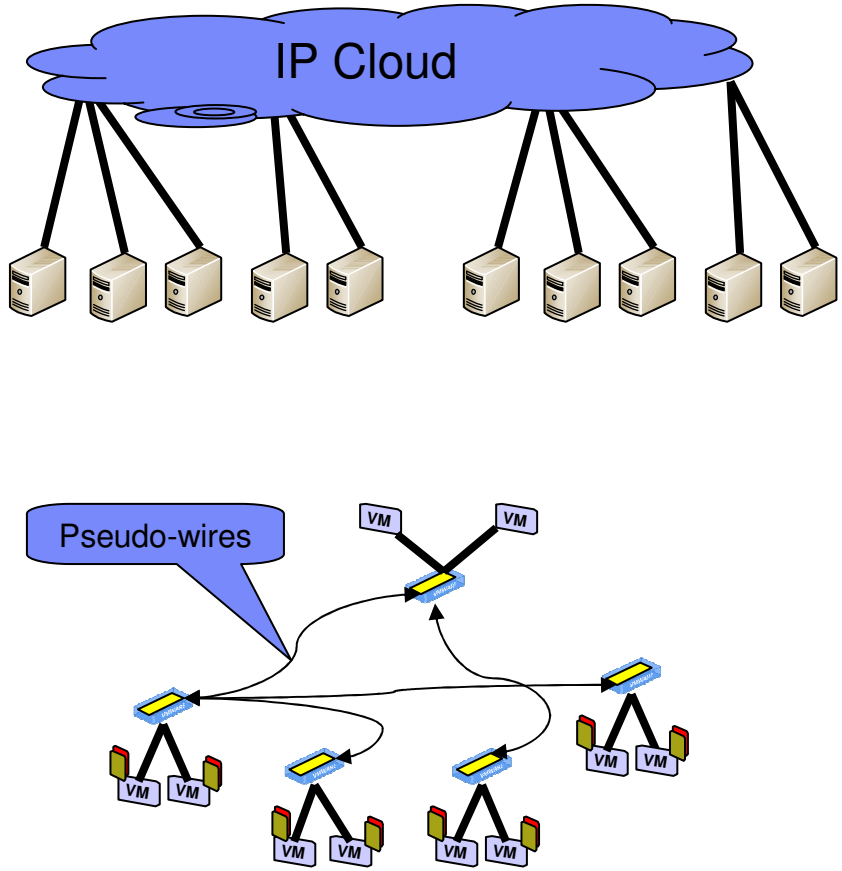


Virtualization – Abstracted Network

- Physical Network View
  - End Stations (Servers) have single IP and MAC in the physical network domain are "dumb" (knows only a default GW)
  - Physical network design is unchanged
  - Same scale as today

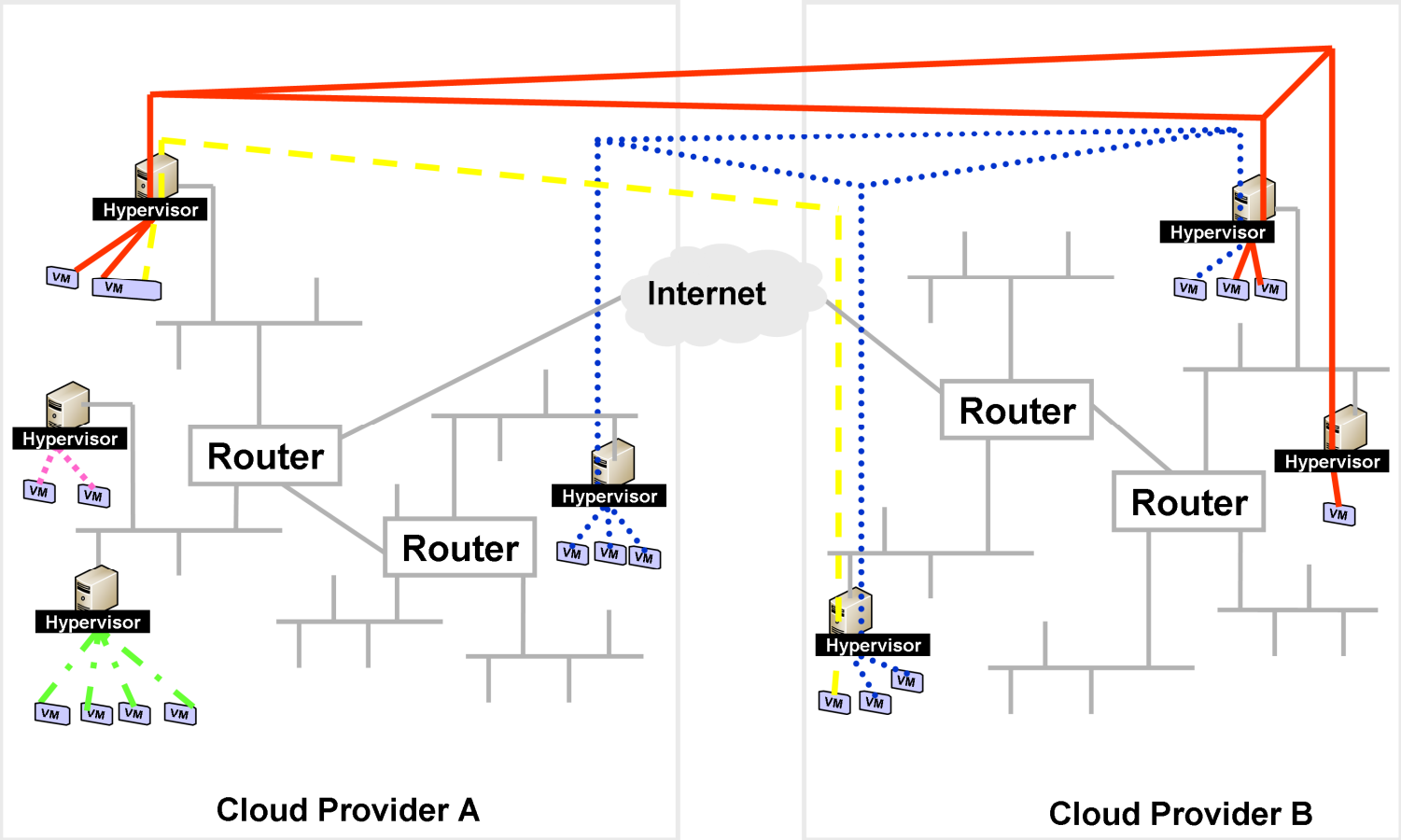
- Abstracted Networks View
  - Hosts hypervisors form an abstracted network between them
  - Abstracted network topology and routes are under the control of the hosts





Virtual Application Networks

- A VAN is a virtual and distributed switching service connecting VMs. VANs revolutionize the way networks are organized by using an overlay network between hypervisors of different hosting platforms.
- The hypervisors' overlay network decouple the virtual network services offered to VMs from the underlying physical network. As a result, the network created between VMs becomes independent from the topology of the physical network.

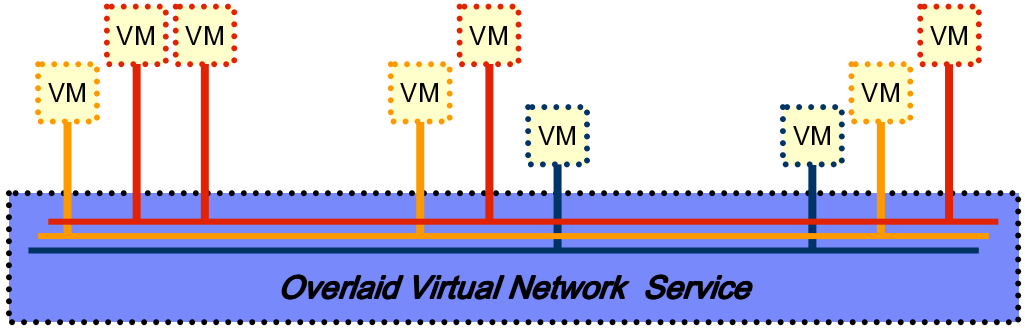


Prior Work:
VIOLIN J. Xuxian, and X. Dongyan (2004); *R. Paul, J. Xuxian, and X. Dongyan (2005)*
VNET: A. I. Sundararaj, and P. A. Dinda (2004); A. I. Sundararaj, A. Gupta, and P. A. Dinda (2004)

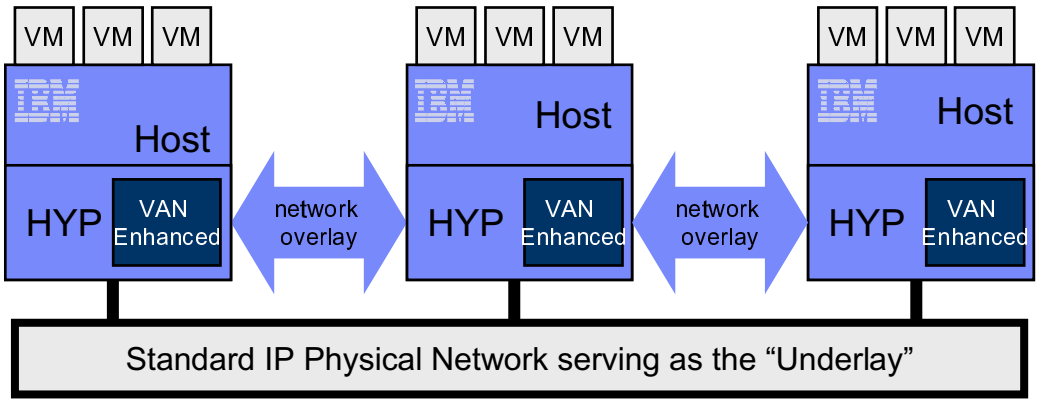
Example for Abstracted Networks

The HRL Virtual Application Networks (VAN) project

- Reservoir Mid 2008-2010
  - An EU project for federated clouds
  - Proof of Concept developed under system X
  - Demonstrated to the EU at EOY-1 and EOY-2
  - Planned demonstration at EOY-3 will includes federated migration

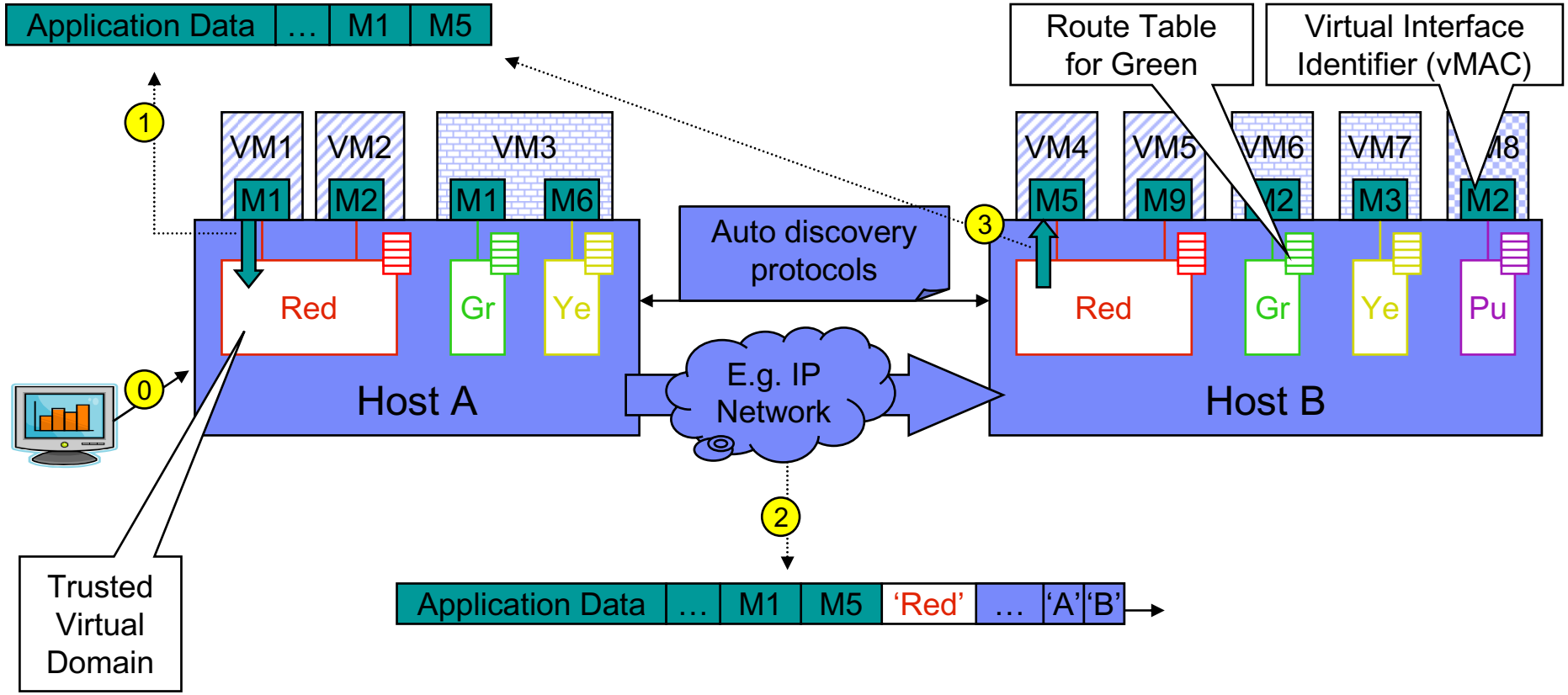


- What are VANs?
  - A design following the Abstracted Network principles
  - A L2-alike service for VMs using IP-based Pseudo-Wires
  - A set of control protocols for setting and maintaining Pseudo-Wires
    - Auto Discovery
    - Dynamic Routing
  - A set of methodologies for using Abstracted Networks



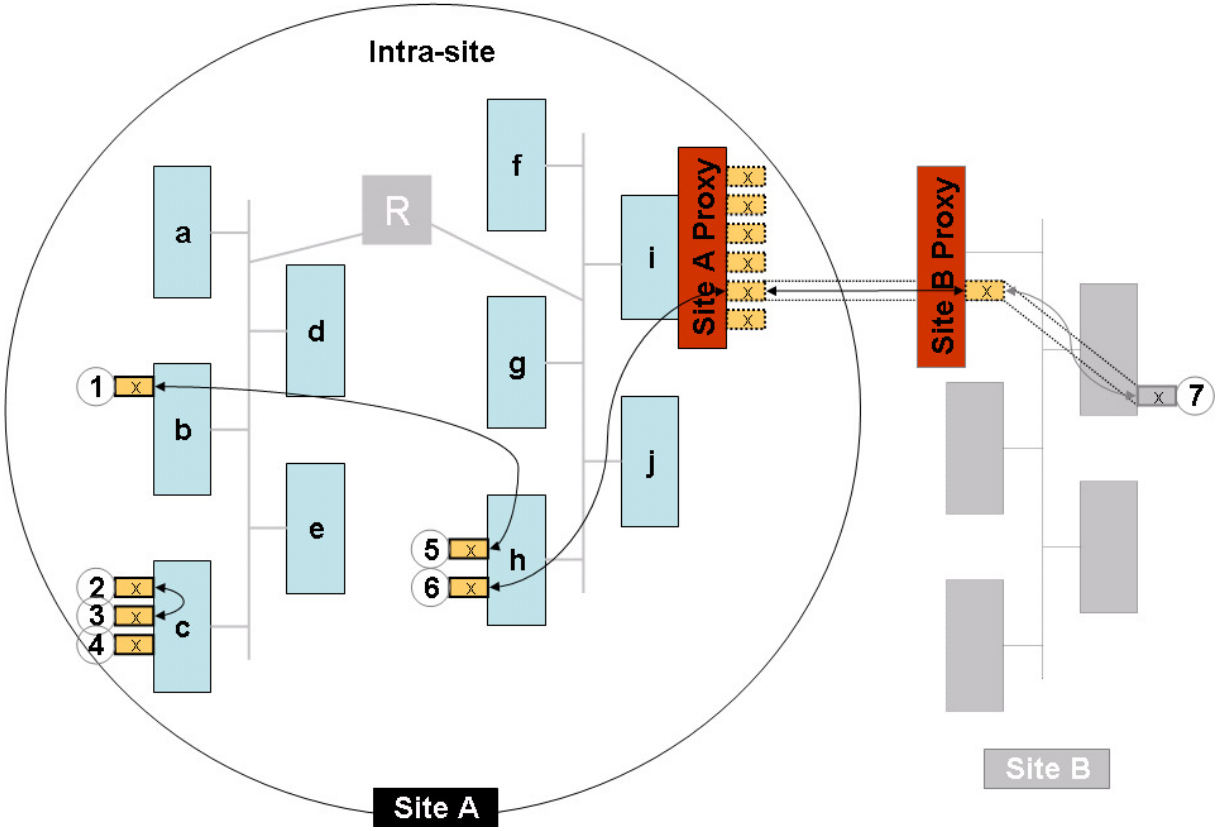
Abstracted Networks Introduce Multiple Research Challenges

- I/O Performance
- Routing
- Traffic shaping
- QoS
- Multi-pathing
- Scalability
- Management (FCAPS)
- Resiliency
- Security
- Network aspects of VM Placement
- Federation of Clouds



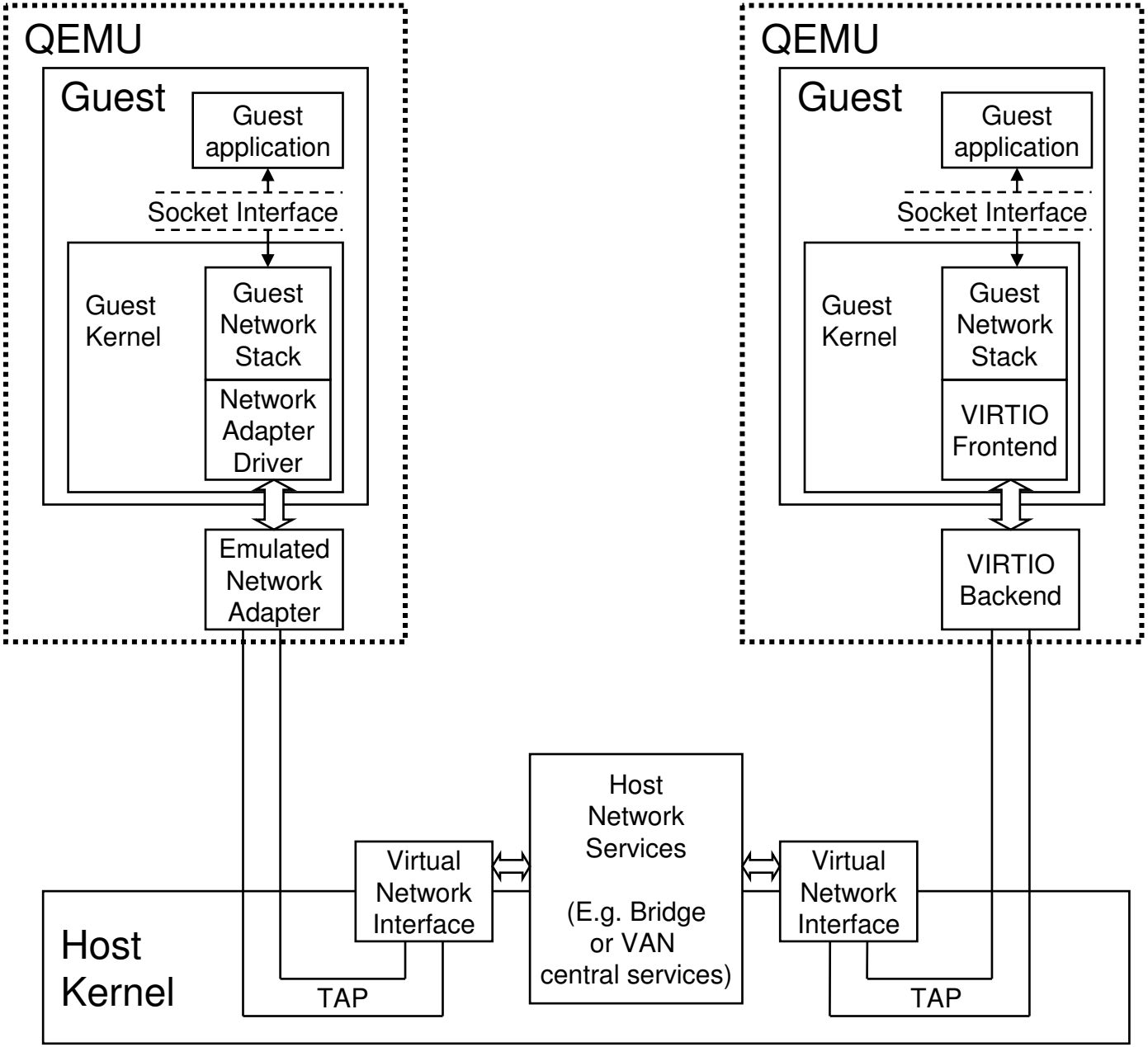
Reservoir – Federated Clouds

- Cloud providers cannot be expected to coordinate their network maintenance, network topologies, etc. with each other.
- A research into Federated Clouds suggests meeting these requirements by separating clouds using VAN proxies, which acts as gateways between clouds.
  - A VAN proxy hides the internal structure of the cloud from other clouds in a federation.
  - The VAN proxies of different clouds communicate to ensure that VANs can extend across a cloud boundary while adhering to the privacy and security limitations of its members

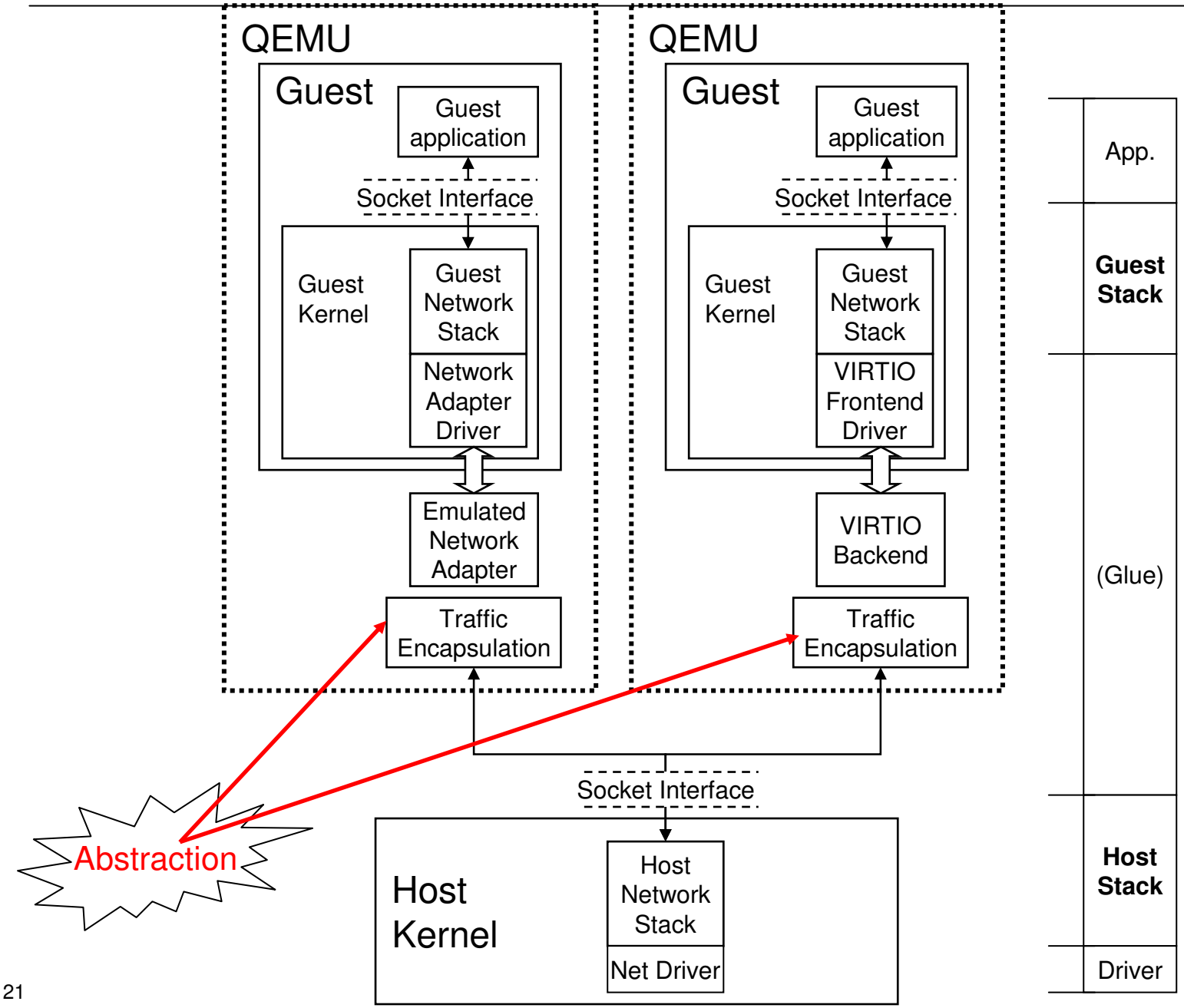


Back

Packet flow today (in KVM)



Pseudo-Wires Between Hosts Results in a Dual Stack



Performance

- Path from guest to wire is long
  - Latencies are manifested in the form of:
    - Packet copies
    - VM exits and entries
    - User/Kernel mode switches
    - Host QEMU process scheduling
- Performance Aspects
  - Increase Packet Size
  - Inhibit Checksum
  - CPU Affinity
  - Flow Control

Increase Packet Size

- Large Packets
  - Transport and Network layers capable of up to 64KB packets
  - Ethernet limit is 1500 bytes but there is no Ethernet wire between guest and host!
  - Set MTU to 64KB in guest

- Flow
  - Application writes 64KB to TCP socket
  - TCP, IP check MTU (=64KB) and create 1 TCP segment, 1 IP packet
  - Guest virtual NIC driver copies entire 64KB frame to host
  - Host writes 64KB frame into UDP socket
  - Host stack creates 1 64KB UDP packet
  - If packet destination = VM on local host
    - Transfer 64KB packet directly on the loopback interface
  - If packet destination = other host
    - Host NIC segments 64KB packet in hardware

Back

Inhibit Checksums

- VM to VM packets represent inner Host buffer transfer
  - Inhibit TCP/UDP checksum calculation and verification
- VM to Network packets are protected by lower stack checksum
  - Inhibit TCP/UDP checksum calculation and verification

Back

CPU affinity and pinning

- QEMU process contains 2 threads
  - CPU thread (actually, one CPU thread per guest vCPU)
  - IO thread
- Linux process scheduler selects core(s) to run threads on
- Many times scheduler made **wrong decisions**
  - Schedule both on same core
  - Constantly reschedule (core 0 -> 1 -> 0 -> 1 -> …)
- Solution/workaround – pin CPU thread to core 0, IO thread to core 1

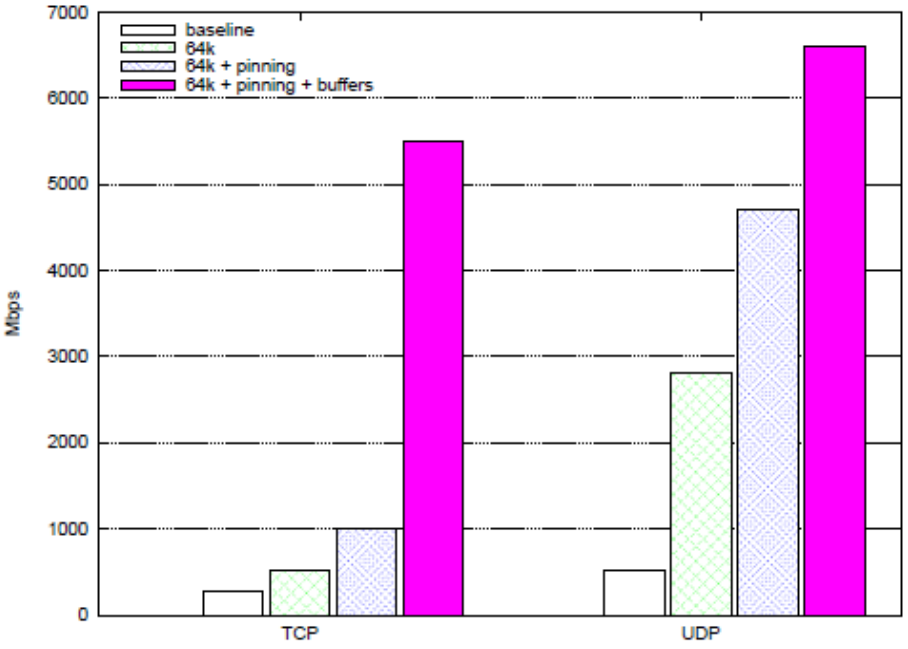
Back

Flow control

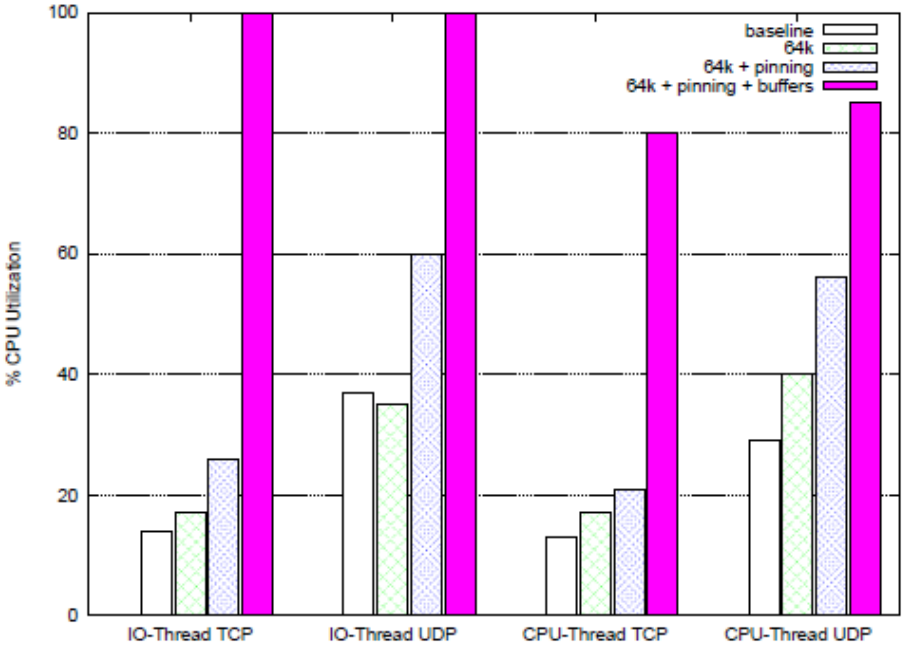
- Guest does not anticipate flow control at Layer-2
- Thus, host should not provide flow control
  - Otherwise, bad effects similar to TCP-in-TCP encapsulation will happen
- Lacking flow control, host should have large enough socket buffers
- Example:
  - Guest uses TCP
  - **Host buffers** should be at least guest TCP's **bandwidth x delay**

Back

Performance results



Throughput



Receiver CPU Utilization

Back