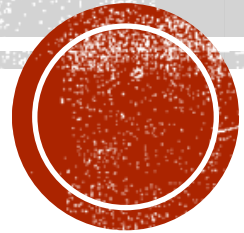
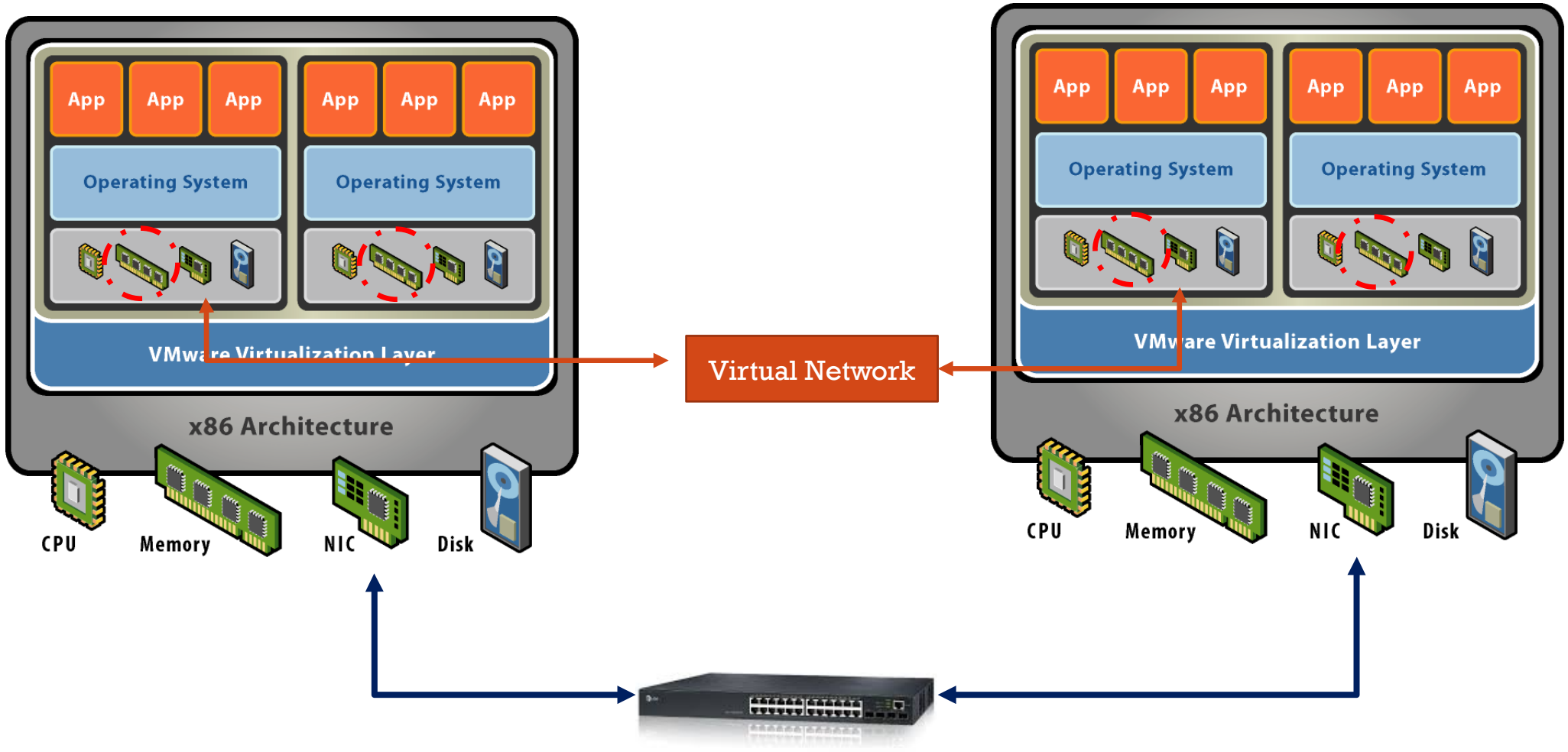
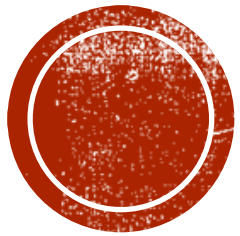
MEMORY & NETWORK VIRTUALIZATION



VIRTUALIZATION





MEMORY VIRTUALIZATION

VIRTUAL MEMORY

Virtual Memory is an automatic address translation technology that provides:

- Decoupling of program's "name space" from physical location
- Provides access to name space potentially greater in size than physical memory
- Expandability of used name space without reallocation of existing memory
- Protection of a given task from interference with its name space by other tasks

VIRTUAL MEMORY COMPONENTS

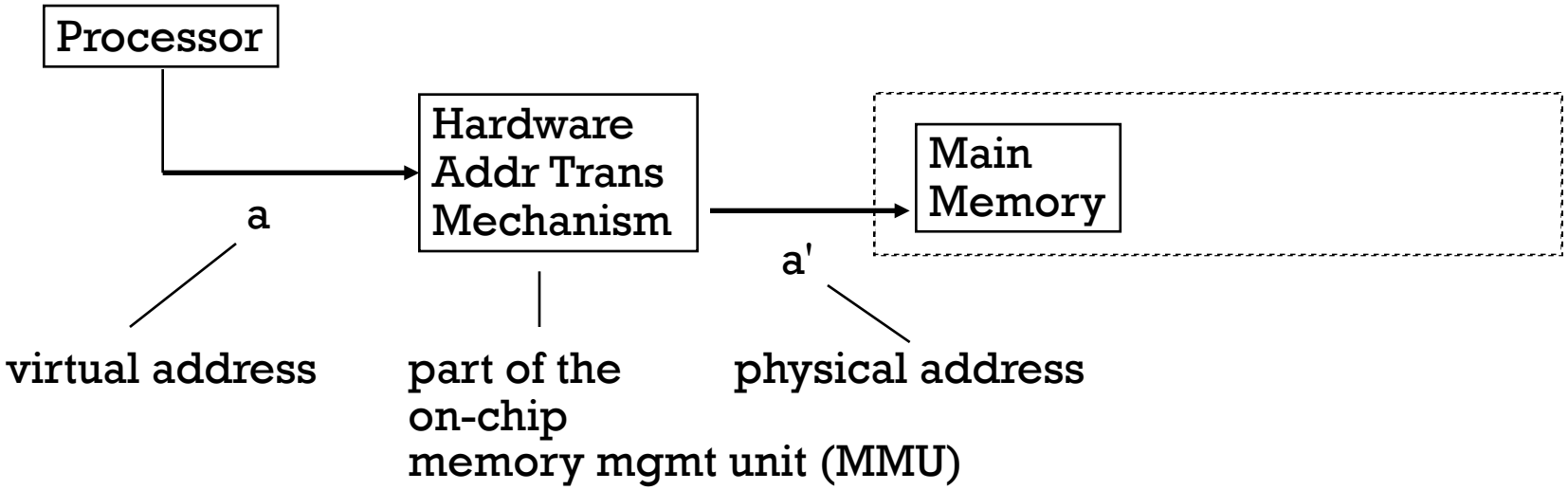
Components that make virtual memory work

- **Physical memory** divided up into pages
- **A swap device**, typically a hard disk, that holds pages not resident in physical memory
  - Also referred to as backing store
- **Address translation**
- **Page tables** to hold virtual-to-physical address mappings
- **Translation Lookaside Buffer** to cache the translation information
- **Management software** in the operating system

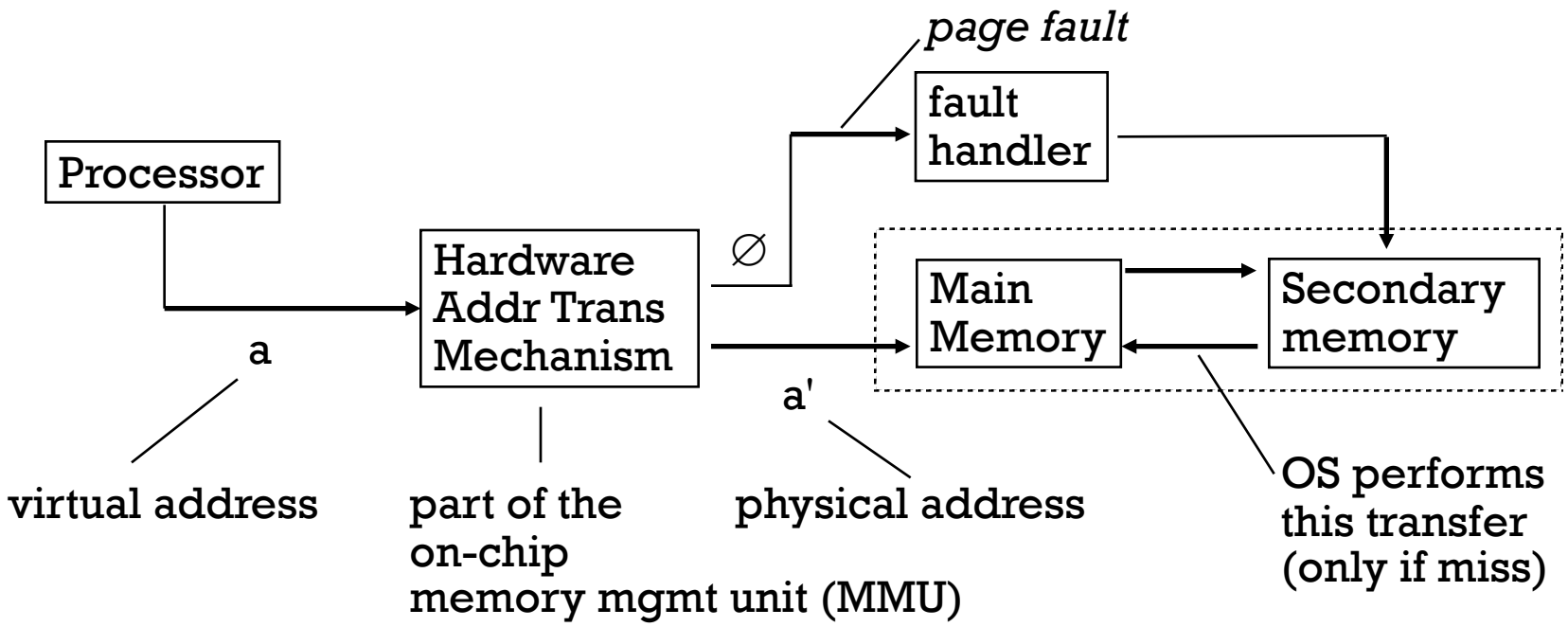
VM ADDRESS TRANSLATION

- Virtual Address Space
  - V = {0, 1, …, N–1}
- Physical Address Space
  - P = {0, 1, …, M–1}
  - M < N
- Address Translation
  - MAP: V $\rightarrow$ P U {$\varnothing$}
  - For virtual address a:
    - MAP(a) = a' if data at virtual address a at physical address a' in P
    - MAP(a) = $\varnothing$ if data at virtual address a not in physical memory
      - Either invalid or stored on disk

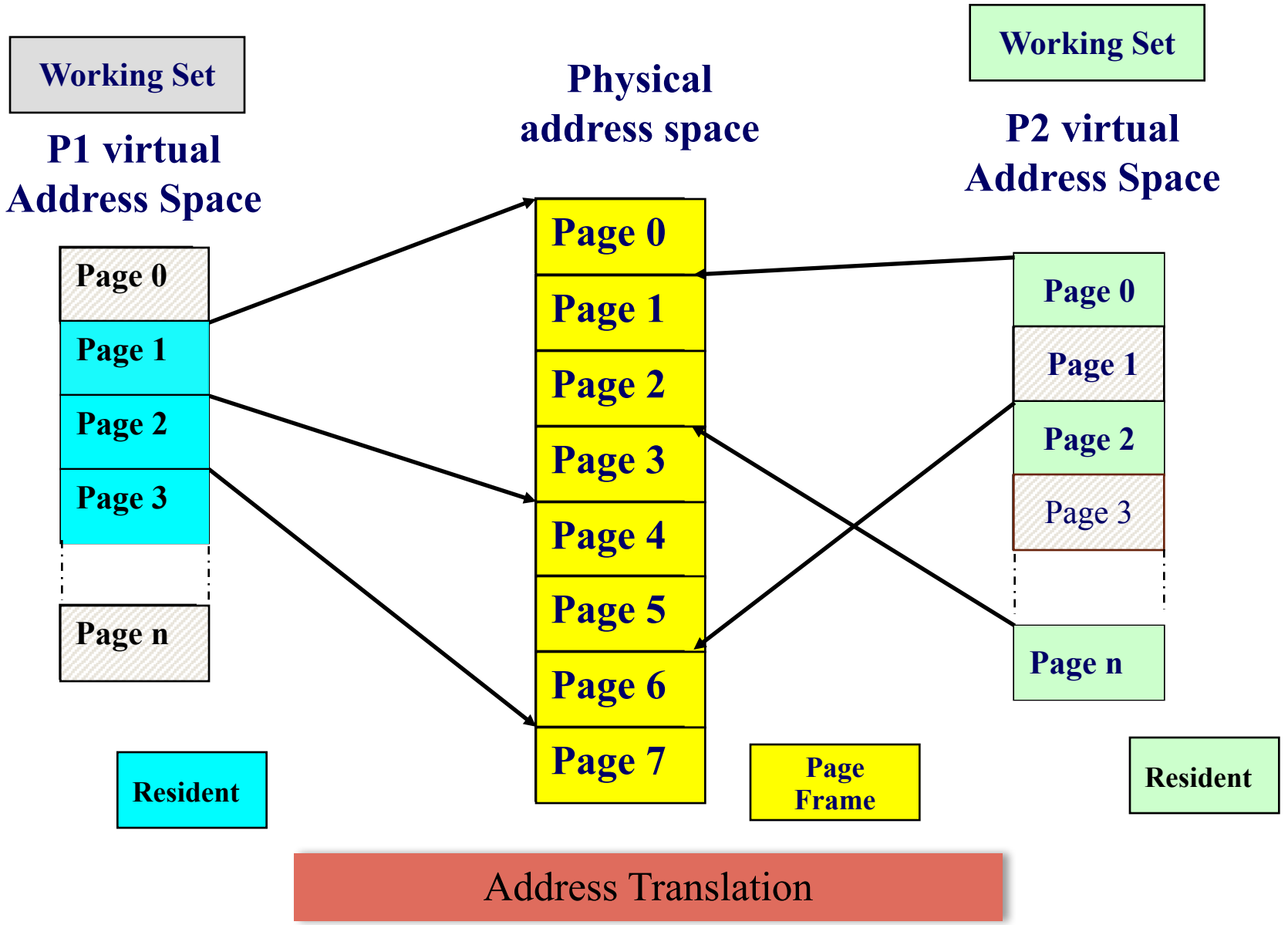
VM ADDRESS TRANSLATION: HIT



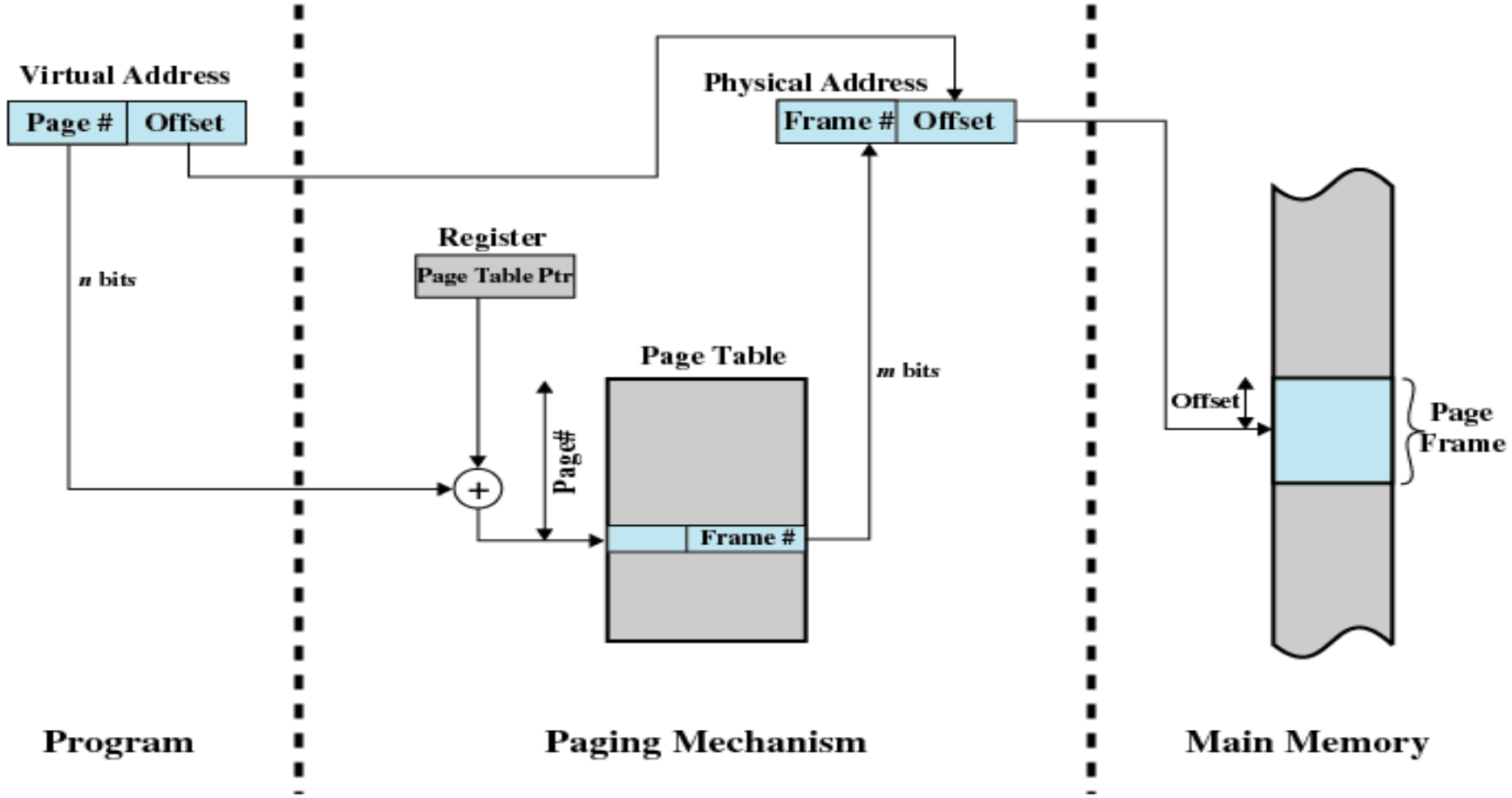
VM ADDRESS TRANSLATION: MISS



PAGED VIRTUAL MEMORY



Address Translation



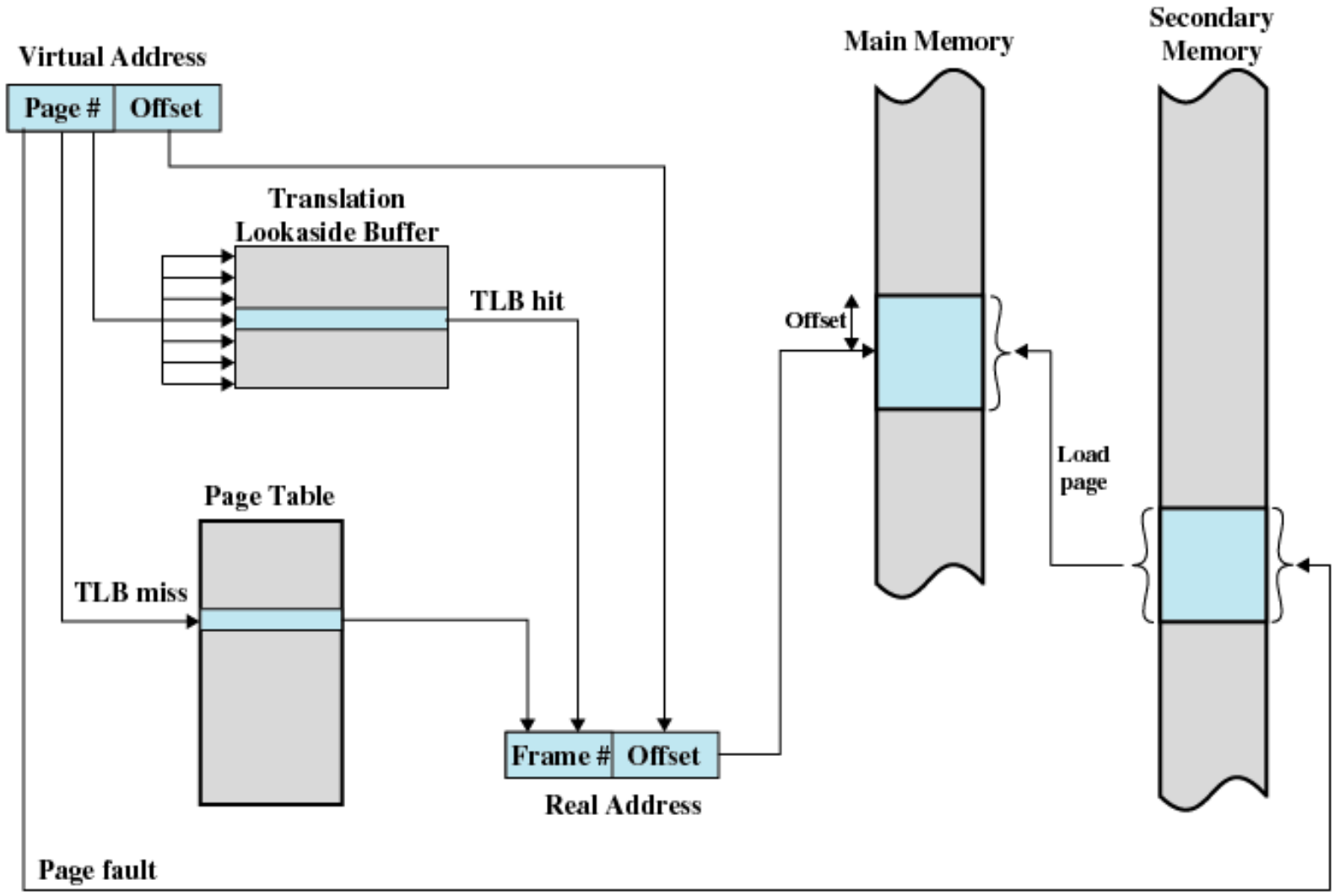
Address Translation

TRANSLATION LOOKASIDE BUFFER

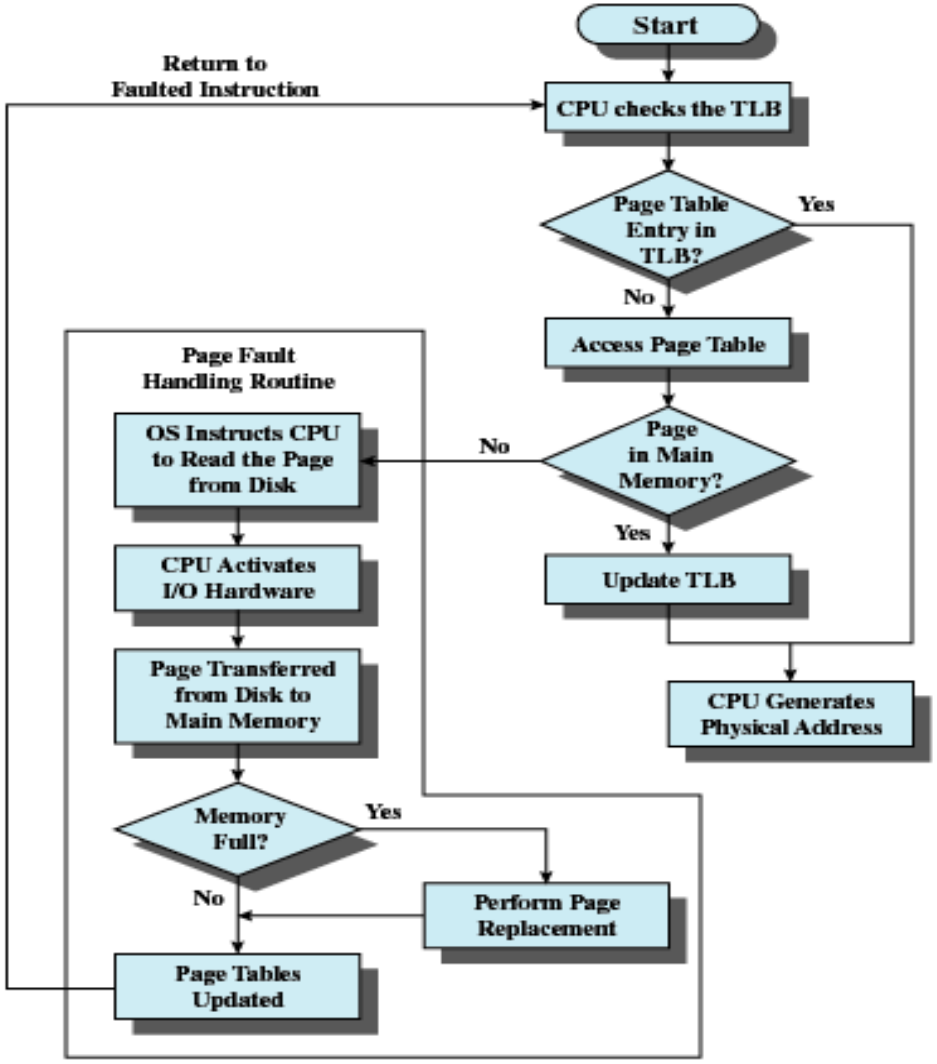
- Each virtual memory reference can cause two physical memory accesses
  - One to fetch the page table
  - One to fetch the data
- To overcome this problem a high-speed cache is set up for page table entries
  - Translation Lookaside Buffer (TLB)
  - Contains page table entries that have been most recently used

TRANSLATION LOOKASIDE BUFFER

- Given a virtual address, processor examines the TLB
- If page table entry is present (TLB hit), the frame number is retrieved and the real address is formed
- If page table entry is not found in the TLB (TLB miss), the page number is used to index the process page table
- First checks if page is already in main memory
  - If not in main memory a page fault is issued
- The TLB is updated to include the new page entry



Translation Lookaside Buffer



Paging and Translation Look Aside Buffer

TLB IN A MULTITASKING ENVIRONMENT

NON-VIRTUALIZED SYSTEM

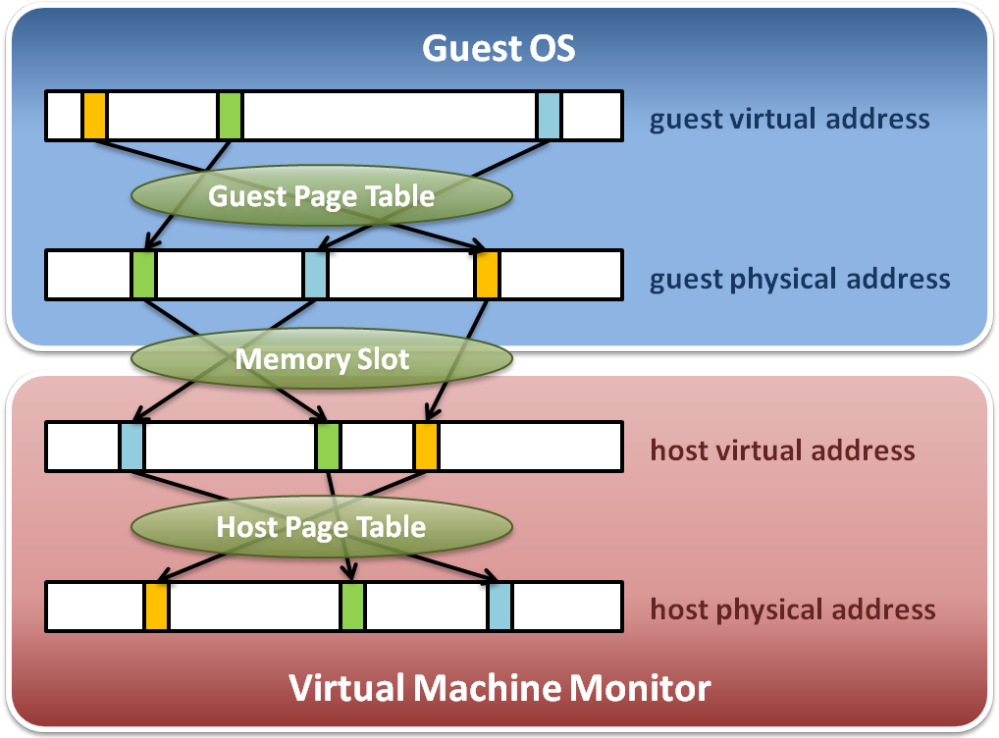
- Most tasks have unique page tables
  - OS must reset the TLB each time it switches from one process to another
  - Flushing and reloading TLB can impact performance, especially if tasks run only for a short amount of time
- To mitigate the impact of task switching, task ID field is added to each TLB
  - This allows the system to retain the mapping information of multiple tasks in the TLB, while switching between tasks
    - Eliminates the need to flush the TLB when switching between tasks

TLB IN A MULTITASKING ENVIRONMENT

VIRTUALIZED SYSTEM

- A guest OS running on virtual machine is unaware of other guests
  - Each OS assigns unique IDs to tasks within its own environment
    - Task IDs across the physical system are no longer guaranteed to be unique
- Trivial solution to the problem

  VM Monitor flushes the TLB every time it switches from one VM to another
  - Forces the task executing in the next VM to reload the TLB with its own page table references
  - Potential for significant performance hit



TLB IN A MULTITASKING ENVIRONMENT

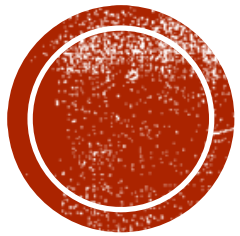
VIRTUALIZED SYSTEM

- Efficient Solution for TLB flushing and reloading problem
  - The concept of tagged translation lookside buffer (tagged TLB)
    - Augment the TLB with VM-specific tag, referred to as *Address Space IDentifier (ASID)*
- Each VM is associated with a unique ASID known only to the hypervisor
  - ASID is invisible to the guest host, thereby eliminating the need to modify original host environment
  - Forces the task executing in the next VM to reload the TLB with its own page table references
    - Improves performance, depending on the workload

TLB IN A MULTITASKING ENVIRONMENT

VIRTUALIZED SYSTEM

- The concept of tagged translation lookside buffer allows support of multiple Virtual Machines, but does not achieve the "OS full control" abstraction expected in a virtualized system
  - Merely providing access to a virtualized system with a virtualized physical memory – Guest physical memory
  - Still the need to map from the virtualized physical memory to the actual physical memory of the host system

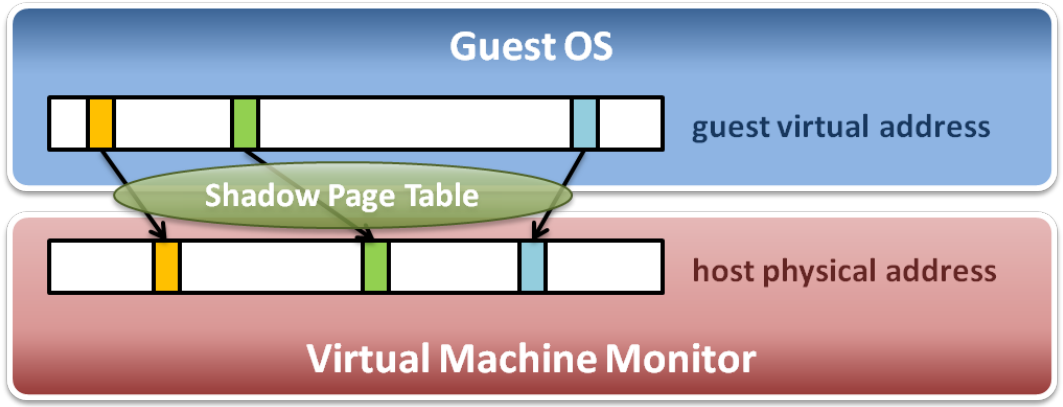


SHADOW PAGE TABLE
MEMORY VIRTUALIZATION FOR VM

SHADOW PAGE TABLES

VMM maintains shadow page tables :

- Direct virtual-to-physical address mapping
- Use hardware TLB for address translation

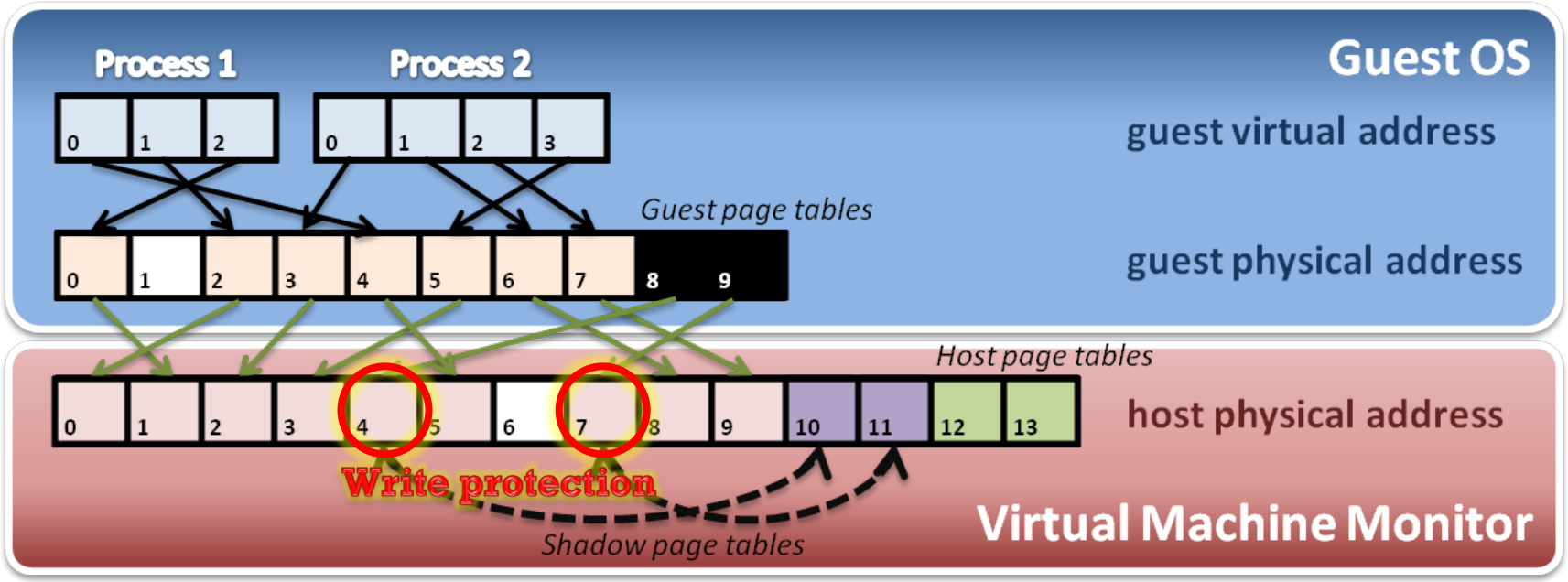


SHADOW PAGE TABLE

- Map guest virtual address to host physical address
  - Shadow page table
    - Guest OS will maintain its own virtual memory page table in the guest physical memory frames.
    - For each guest physical memory frame, VMM should map it to host physical memory frame.
    - Shadow page table maintains the mapping from guest virtual address to host physical address.
  - Page table protection
    - VMM will apply write protection to all the physical frames of guest page tables, which lead the guest page table write exception and trap to VMM.

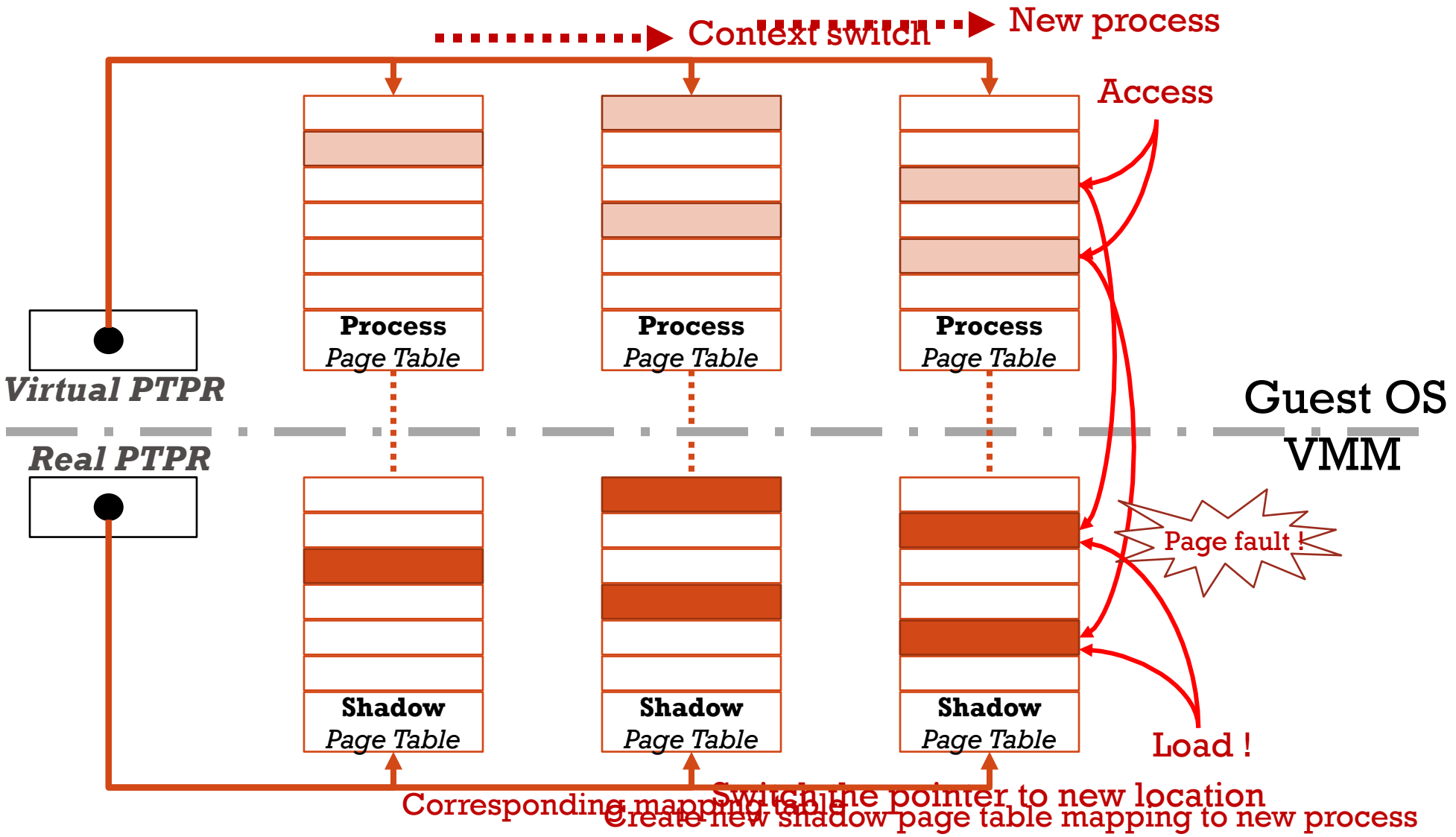
SHADOW PAGE TABLE

- Construct shadow page table
  - Guest OS will maintain its own page table for each process.
  - VMM maps each guest physical page to host physical page.
  - Create shadow page tables for each guest page table.
  - VMM should protect host frame which contains guest page table.



SHADOW PAGE TABLE

- Shadow page table operations :



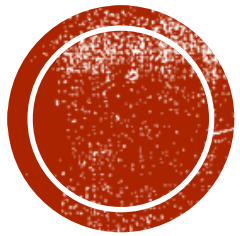
SHADOW PAGE TABLES LIMITATIONS

- Maintaining consistency between guest page tables and shadow page tables leads to an overhead
  - It generates VMM traps
- Memory overhead due to shadow copying of guest page tables
- Shadow page table implementation is extremely complex
- Page fault mechanism and synchronization issues are critical

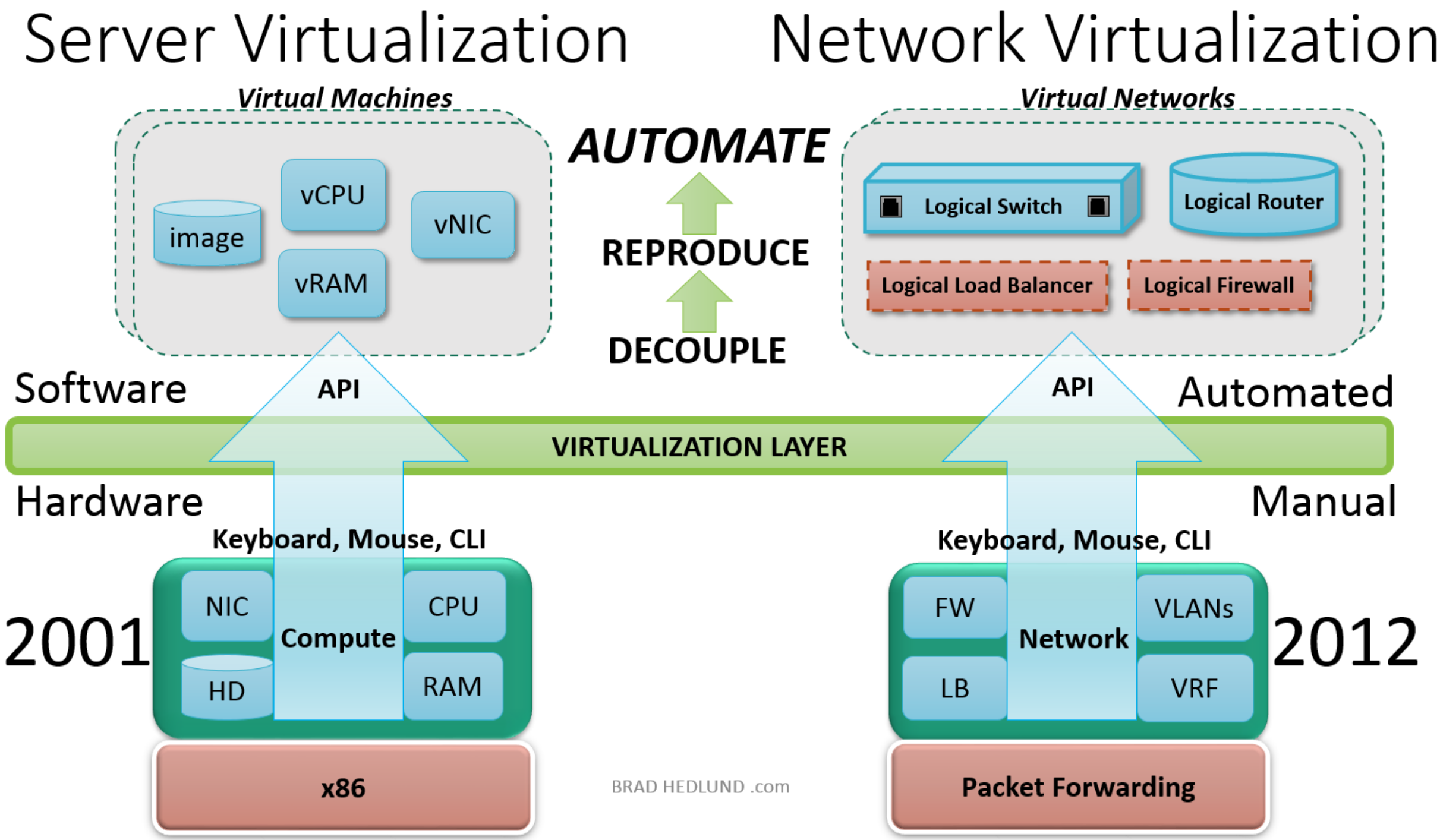
HARDWARE ASSISTANCE MEMORY VIRTUALIZATION

VIRTUALIZED SYSTEM

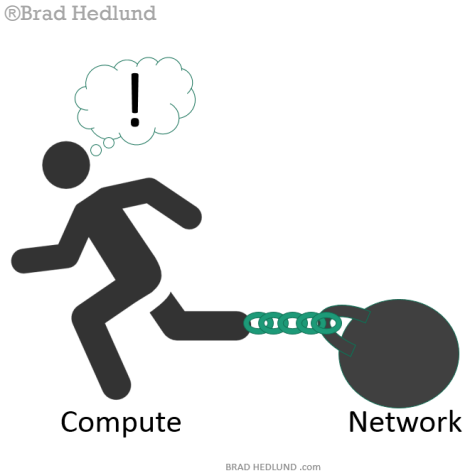
- Unlike in the case of Shadow Tables, where the mapping is carried out in Software, Virtual Page Tables mapping is built directly into the CPU
  - Significant performance improvement
- Modern CPU architectures maintain a guest TLB, where Virtual Page Table translations are stored



NETWORK VIRTUALIZATION


Server Virtualization
Network Virtualization
Virtual Machines
Virtual Networks
AUTOMATE
REPRODUCE
DECOUPLE
image
vCPU
vNIC
vRAM
Logical Switch
Logical Router
Logical Load Balancer
Logical Firewall
Software
API
API
Automated
VIRTUALIZATION LAYER
Hardware
Manual
Keyboard, Mouse, CLI
Keyboard, Mouse, CLI
2001
NIC
CPU
Compute
HD
RAM
FW
VLANs
Network
LB
VRF
2012
x86
Packet Forwarding
BRAD HEDLUND .com

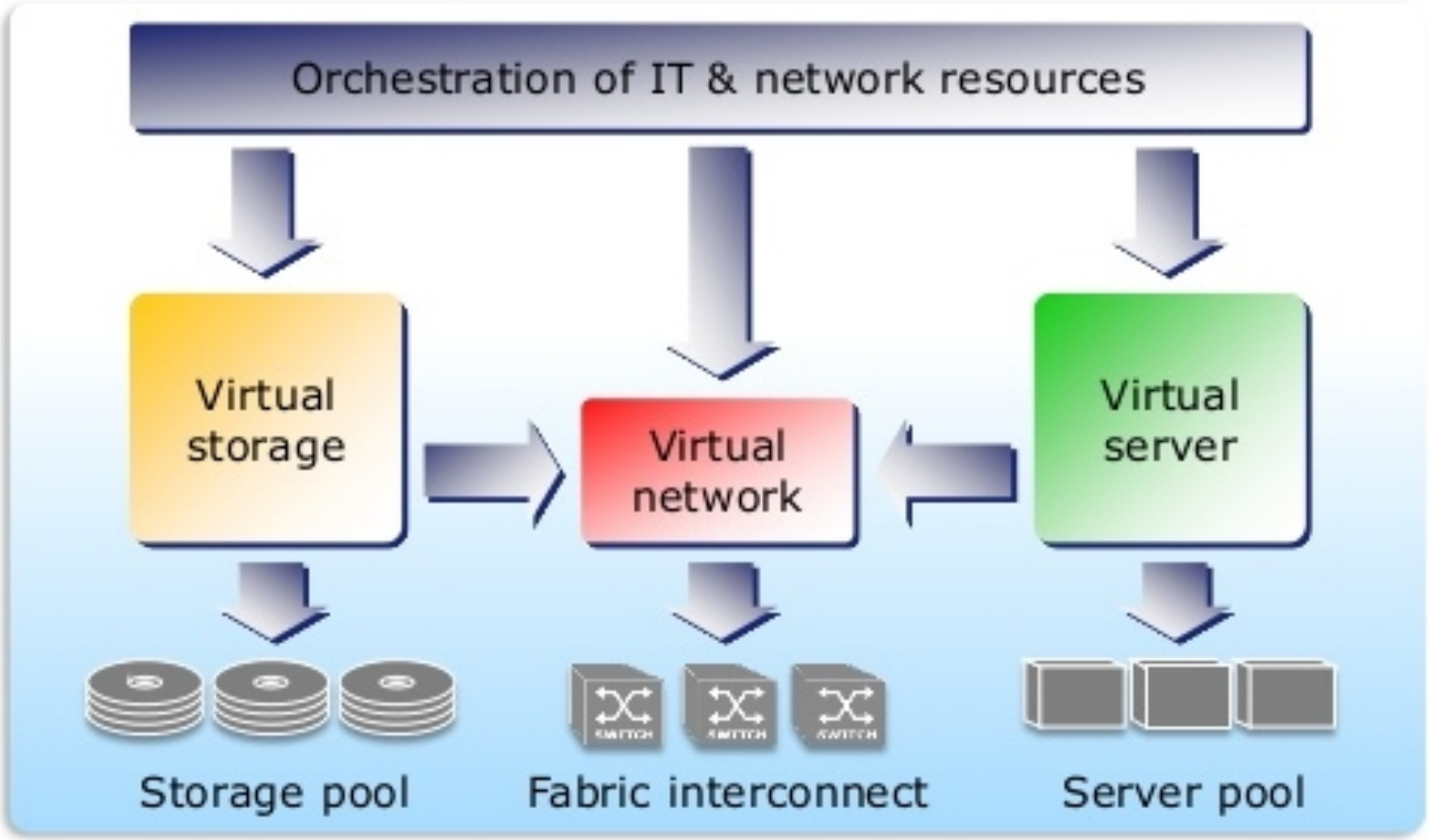
ALL INFRASTRUCTURE ORCHESTRATION



PAST

Virtualization requires:

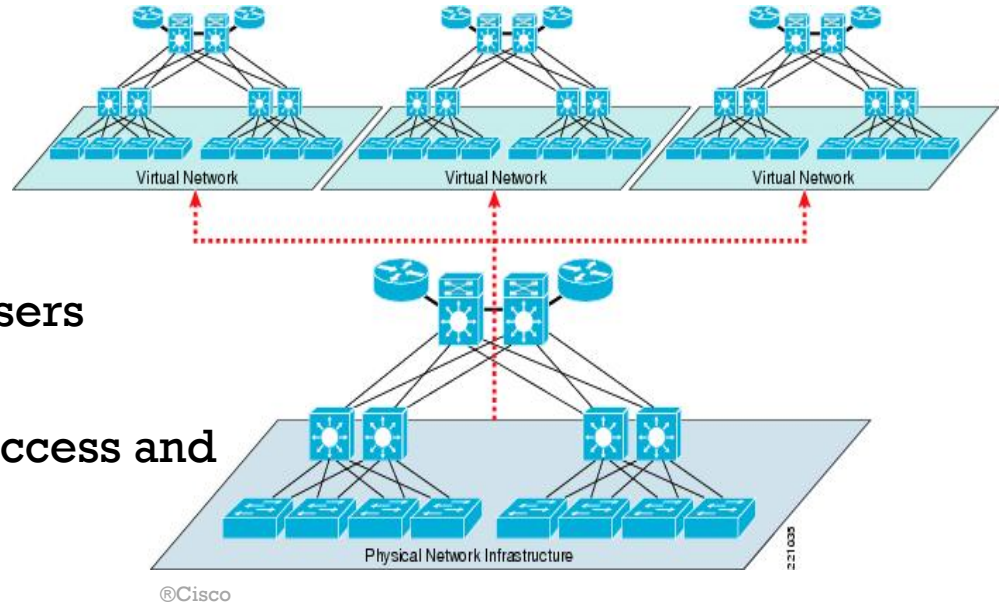
- Lifecycle
- New Identity
- Any Location
- Simple Configuration



NOW & FUTURE

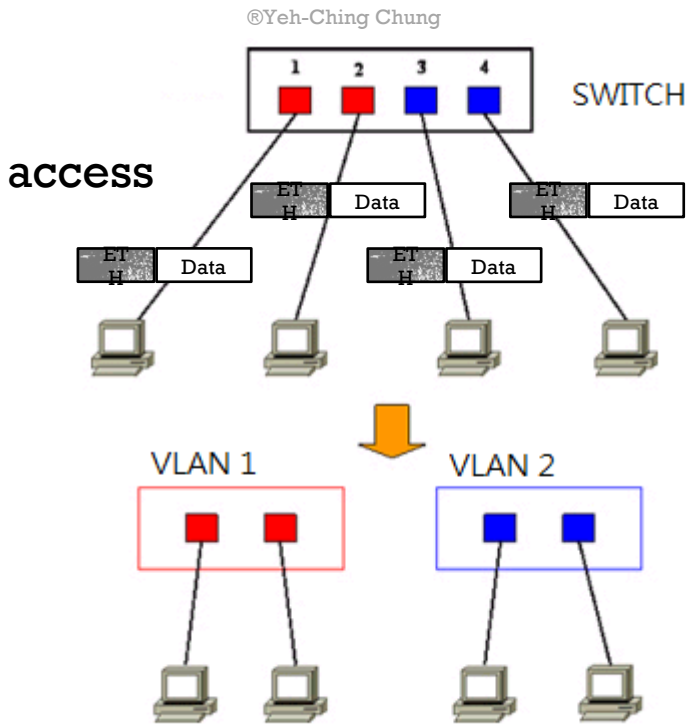
VIRTUAL NETWORK

- **Abstracted network view** for an user
  - Decoupled from physical infrastructure
  - Composed as a set of logical network resources
- Provide **isolation** by:
  - **Address** space - remove the threat of address conflict
  - **Performance** - virtual networking more predictable for users
  - **Management** - mimic usage of non-virtualized network
  - **Security** – don't allow tenant's users (and their traffic) to access and interrupt the work of other tenants
- Configuration **independence** and elasticity
- Easier to **deploy and manage** network services and underlying network resources

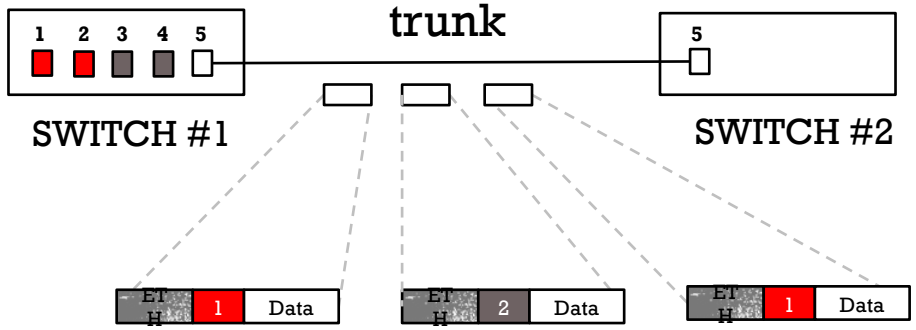


VLAN (VIRTUAL LOCAL AREA NETWORK)
INFRASTRUCTURE SHARING

- Device virtualization
  - Divide physical switch into multiple logical switches

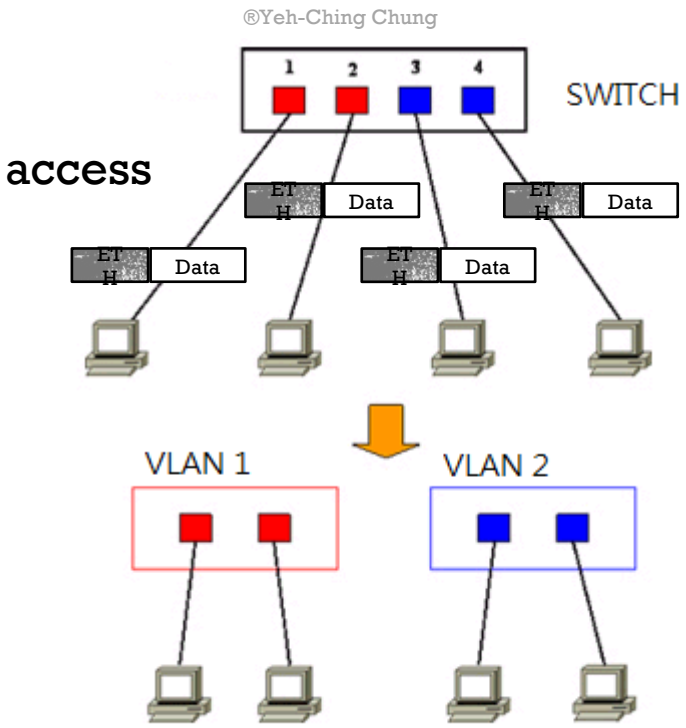


- Link virtualization
  - Divide physical link into multiple logical links



VLAN: RESOURCE SHARING

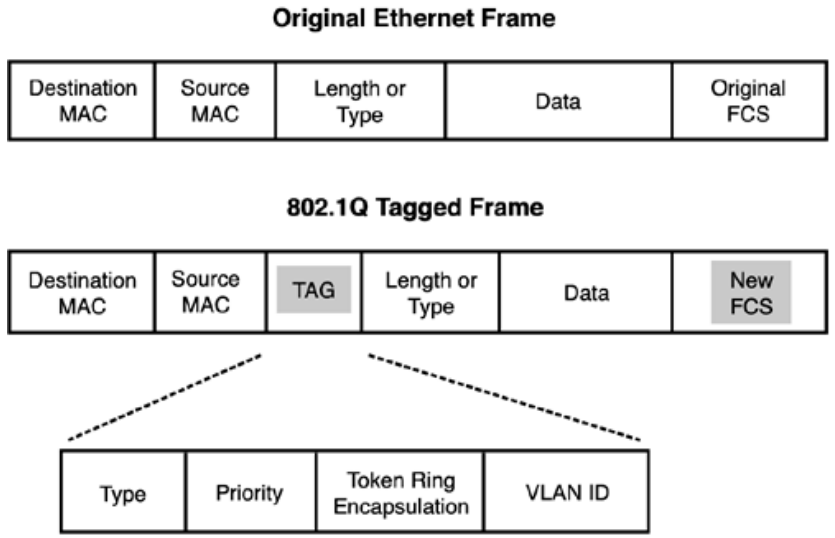
- Device virtualization
  - Divide physical switch into multiple logical switches



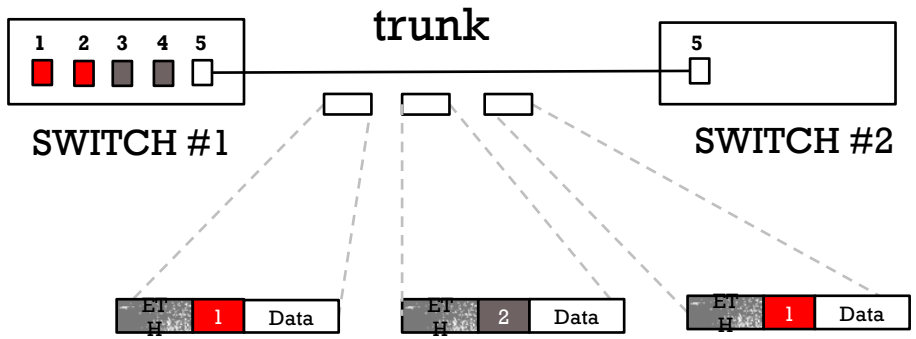
- Virtualization is implemented within switch management software
  - VLAN can be a group of ports
  - VLAN can be group of MAC addresses
  - VLAN can be a specific upper layer protocol
  - VLAN can be a group of IP addresses
  - VLAN can be a group of authenticated users
- A network chip (frame forwarding silicon) is shared by all virtual switches
- Network chip must support VLAN framing and processing

VLAN: LINK VIRTUALIZATION

- Link virtualization is done by network protocol (new Ethernet header 802.1Q)
- Ethernet frame contains new fields
- Link bandwidth is shared between VLANs



- Link virtualization
  - Divide physical link into multiple logical links



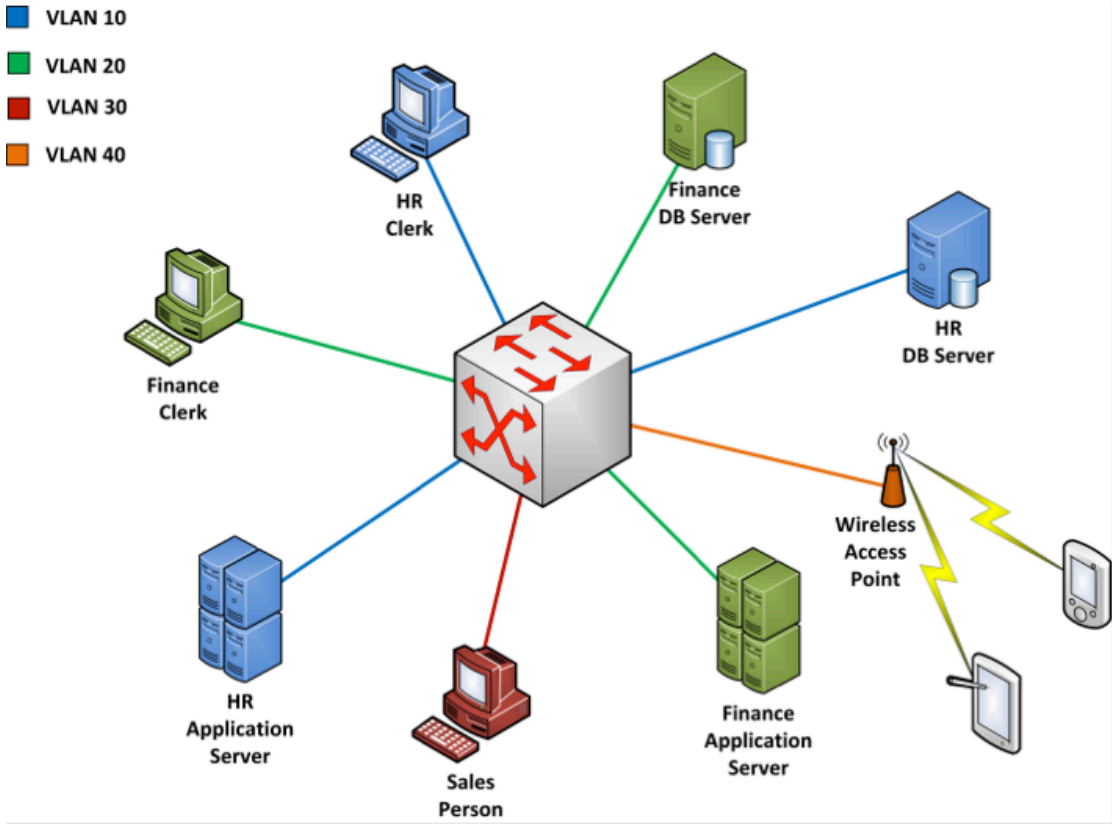
- Virtual links can be isolated one from each other by setting rate limitation per vlan

VLANS IN ENTERPRISES

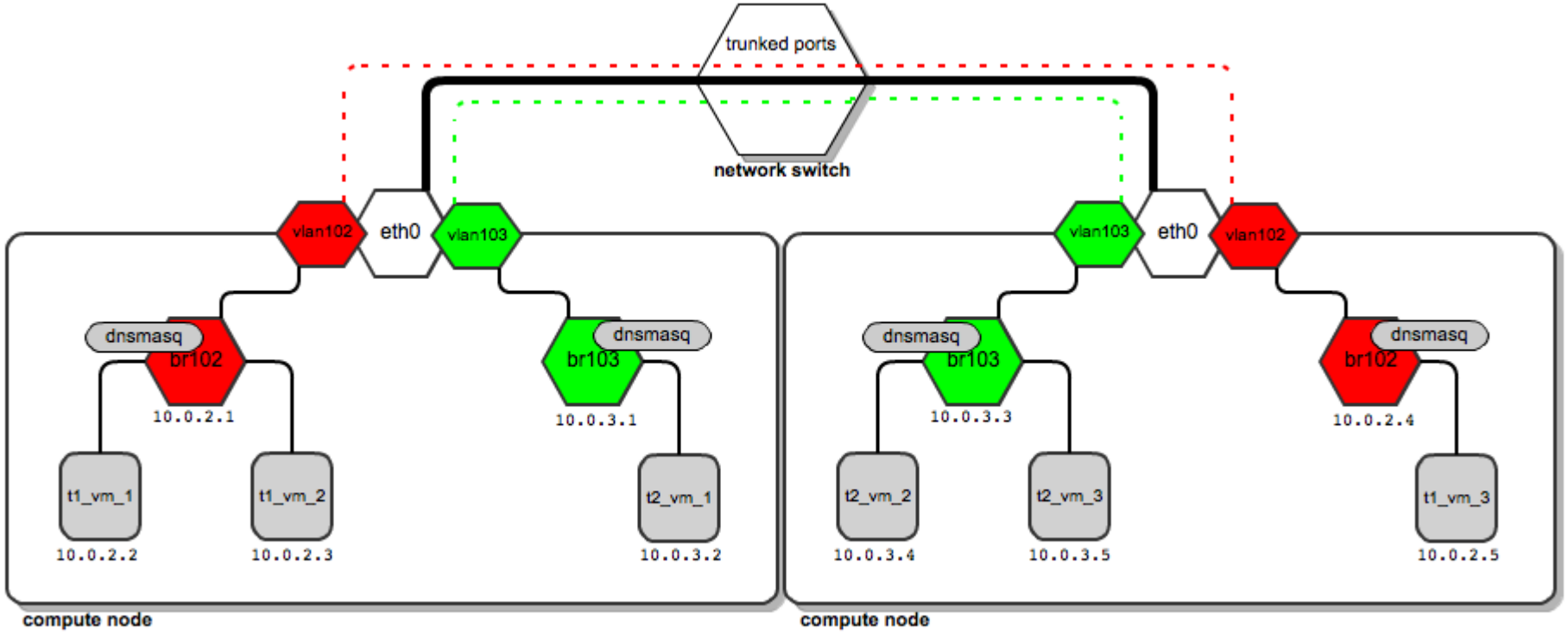
- Grouping devices by organizational/location issues
  - Logical separation between **groups in the organization**
  - VLAN for each **building** or each **floor** of a building

- Grouping devices for security
  - It is often a good practice to put **servers** and key **infrastructure** in their own VLAN, isolating them from the **general broadcast traffic** and enabling greater protection,
  - Any **sensitive data** (financial, research) should have its own VLAN

- Grouping devices by traffic types
  - **VoIP quality** is improved by isolating VoIP devices to their own VLAN.
  - Other traffic types may also warrant their own VLAN:
    - Network **management** traffic
    - IP multicast traffic such as **video**
    - **File and print** services
    - **Email** & Internet **browsing**
    - **Database** access

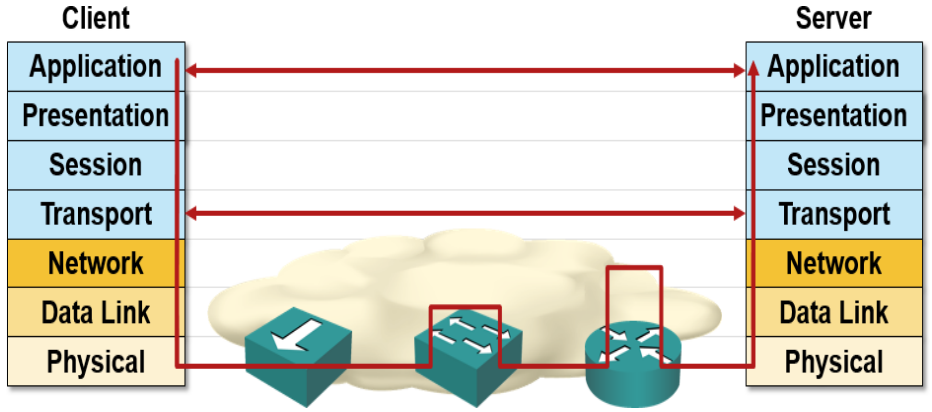


VLAN IN SMALL CLOUD



VLAN LIMITATIONS

It is OSI Layer 2 (Data link)



- Flooding of broadcast frames
  - Every broadcast frame flooded throughout a L2 domain must be processed by every host participating in that domain
  - Every virtualization hypervisor host has to processes every broadcast frame generated anywhere (regardless of whether its VMs belong to the VLAN generating the flood or not)
  - Once you get a loop in a bridged network your network is toast
    - The whole Layer 2 network is a single failure domain
- It uses MAC address which lacks of addressing hierarchy

VLAN FOR CLOUD?

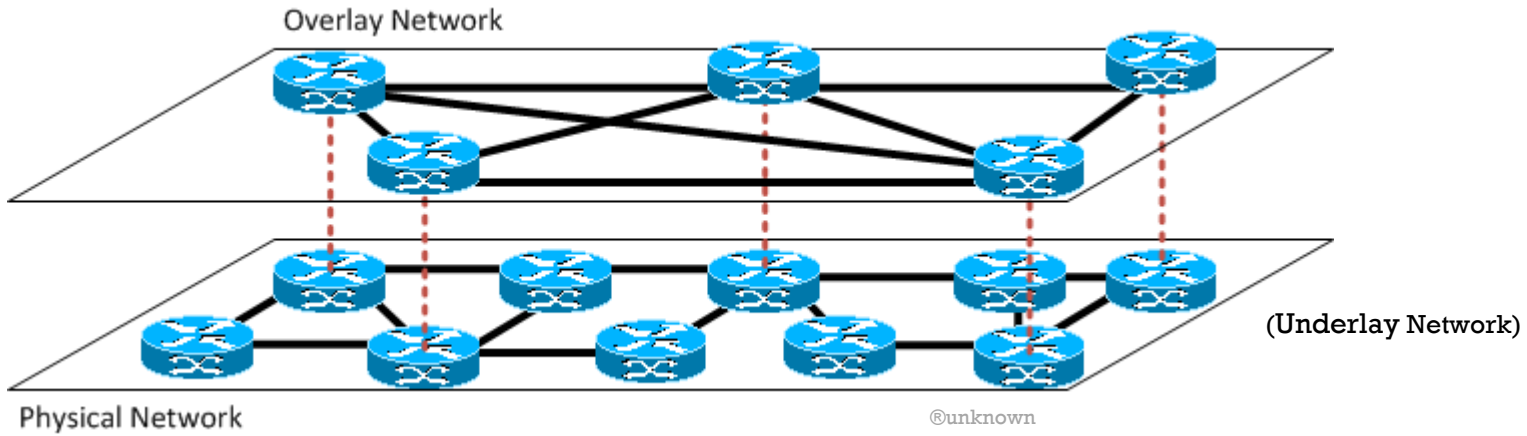
- VLAN advantages
  - **Cheap** in terms of protocol overhead:
    - VLAN tag is only additional 4 bytes of the frame header
  - **Supported** by most of the network devices

- VLAN disadvantages
  - **Not scalable**
    - Only 4096 virtual networks in 802.1Q (vlan_id is 12-bit field)
    - Only 1000 hosts in a virtual network
    - 802.1ad doesn't solve all problems
  - **Management can become complex**
    - To be configured on each device
    - VLAN swapping required if somewhere VLAN tag already used
    - Broadcast storms in case of switching loops affects all VLANs

NETWORK VIRTUALIZATION VIA OVERLAY NETWORK

Overlay networking:

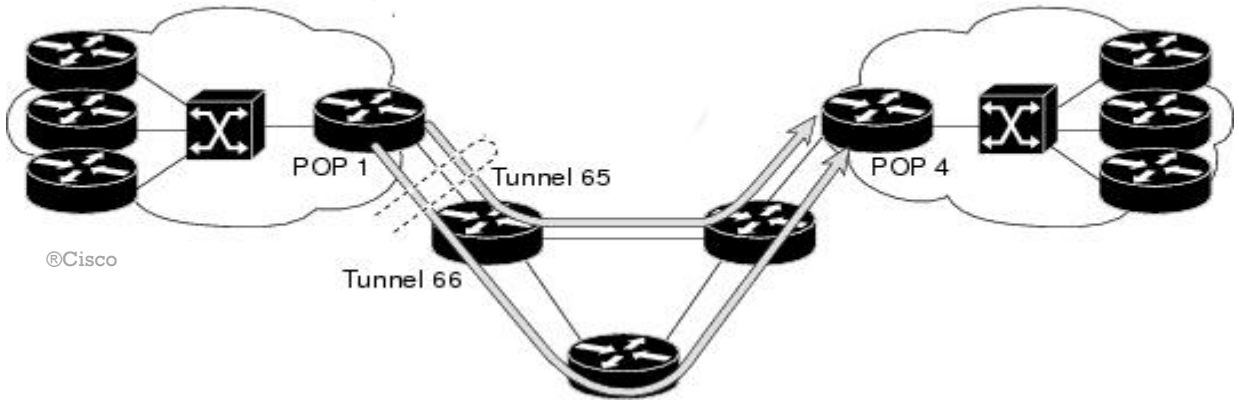
- A virtual network that is **built on top** of an existing physical network (underlay network)
- **Edge nodes** of physical network become nodes of overlay network
- **Tunnels** between edge nodes become logical links of overlay network
- Virtual networking like **yet another network application** (like E-mail, Web, Skype)
- Many virtual networks can **coexist** independently over the same physical network



(Underlay Network)

VIRTUALIZATION TECHNIQUE: TUNNELING

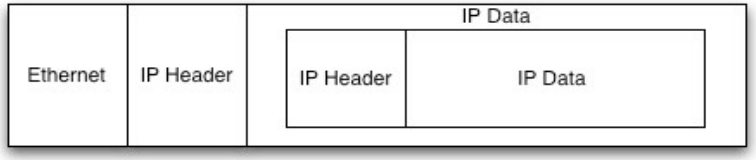
- **Tunnel** is a connection across a network which **ships protocol frames** at payload that normally wouldn't forwarded by network because of breaking of the classical network layering
- Intermediate nodes of tunnel **don’t see encapsulated** frames (it is just data)
- Encapsulated frames could be **encrypted** (SSL/TLS, SSH, IPsec)
- Connecting **distance sites**:
  - Tunnels via global Internet
  - Tunnels via WAN networks



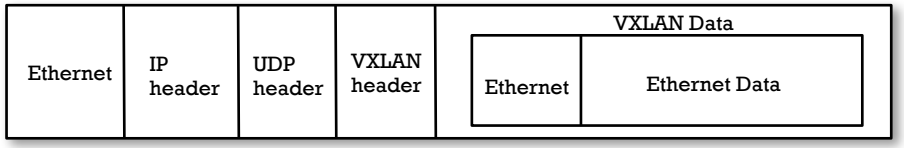
TUNNELING ENCAPSULATION

- Tunneling encapsulation examples:

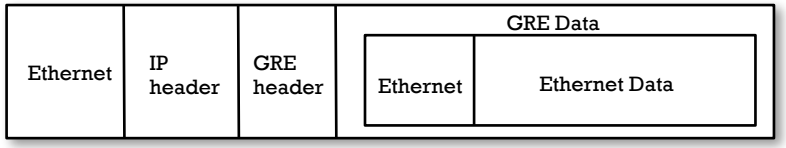
IP in IP



Ethernet in IP (VXLAN)

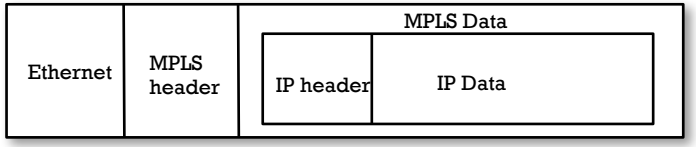


Ethernet in IP (GRE)



Tunnels via Internet

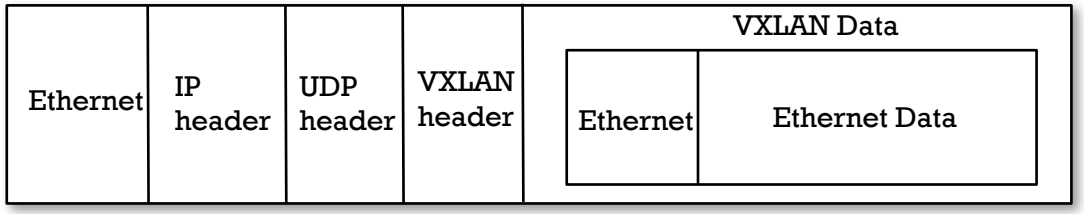
IP in MPLS



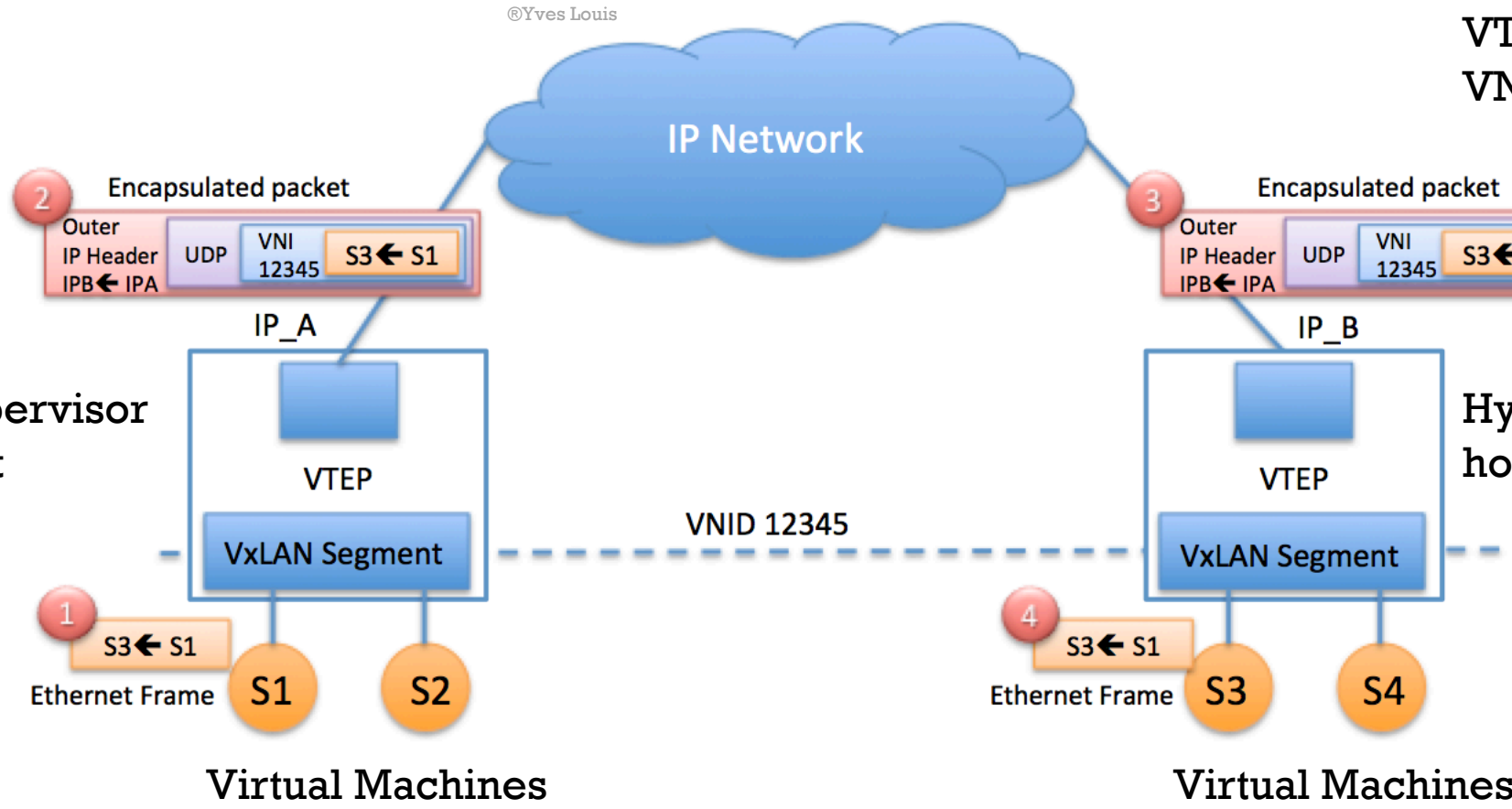
Tunnel via MPLS network (popular service offered by core/ISP networks)

VM OVERLAY NETWORK: VXLAN

- **VXLAN (Virtual Extensible LAN)** – Ethernet over IP
  - **16 millions** logical networks (Layer 2 networks)
    - VNID (VxLAN segment identifier): 24 bits
  - Ethernet **broadcast** domain tunneled across IP network
    - Ethernet broadcast/multicast implemented using IP multicast
  - 50-bytes **overhead** (requires jumbo frames and higher MTU)
  - Virtual Machines **don't aware** of VXLAN usage
  - Hypervisor hosts appear as **simple IP hosts** to the transport network



VM OVERLAY NETWORK: VXLAN



VTEP – Virtual Tunnel End-Point
VNID - VxLAN segment identifier

S1-S4 VMs use Ethernet MAC for frame addressing

It is de facto Ethernet (VMs) over IP (network).

CLOUD OVERLAY NETWORK

- Overlay advantages
  - Full **address isolation** between virtual network and physical underlay infrastructure
  - **Independence** from type of underlay network and its topology:
    - Use existing IP networks and global Internet
    - With additional encapsulation ISP MPLS networks can be also used
  - **No changes** in underlay network – all virtualization **complexity at edges** of network (follows original Internet design)
  - Network **resilience** is provided by underlay network
  - Fair **scalability**
  - Support easy VM **migration** (including policy, security and VLANs)

- Overlay disadvantages
  - Requires **jumbo frames** everywhere:
    - **Wrong MTU** causes problems difficult to be correctly identified and localized
  - Encapsulation introduce **CPU and latency overheads** (up to 60%) due to missing checksum and TCP segmentation offloading
  - Requires **non-oversubscribed** physical underlay network:
    - IP network provide **no throughput isolation** of virtual networks
  - Control Plane **bottleneck** still exists
  - **Gateways** between virtual network and other network may need to pass high volumes of traffic
  - Some **value-added features** in existing networks cannot be leveraged due to encapsulation
    - Traffic engineering in IP core not possible
  - Currently **a lot of solutions** and protocols for creating overlays (compatibility problems)

SUMMARY

- Memory Virtualization
  - Virtual Memory Background
  - Shadow Page Tables: Virtual Memory for VM
- Network Virtualization
  - Existing Network Virtualization: VLAN
  - Overlay Network