z/VM Virtual Switch

20th Edition

Alan Altmark
Senior z/VM Engineer and Consultant
IBM Infrastructure

Alan_Altmark@us.ibm.com

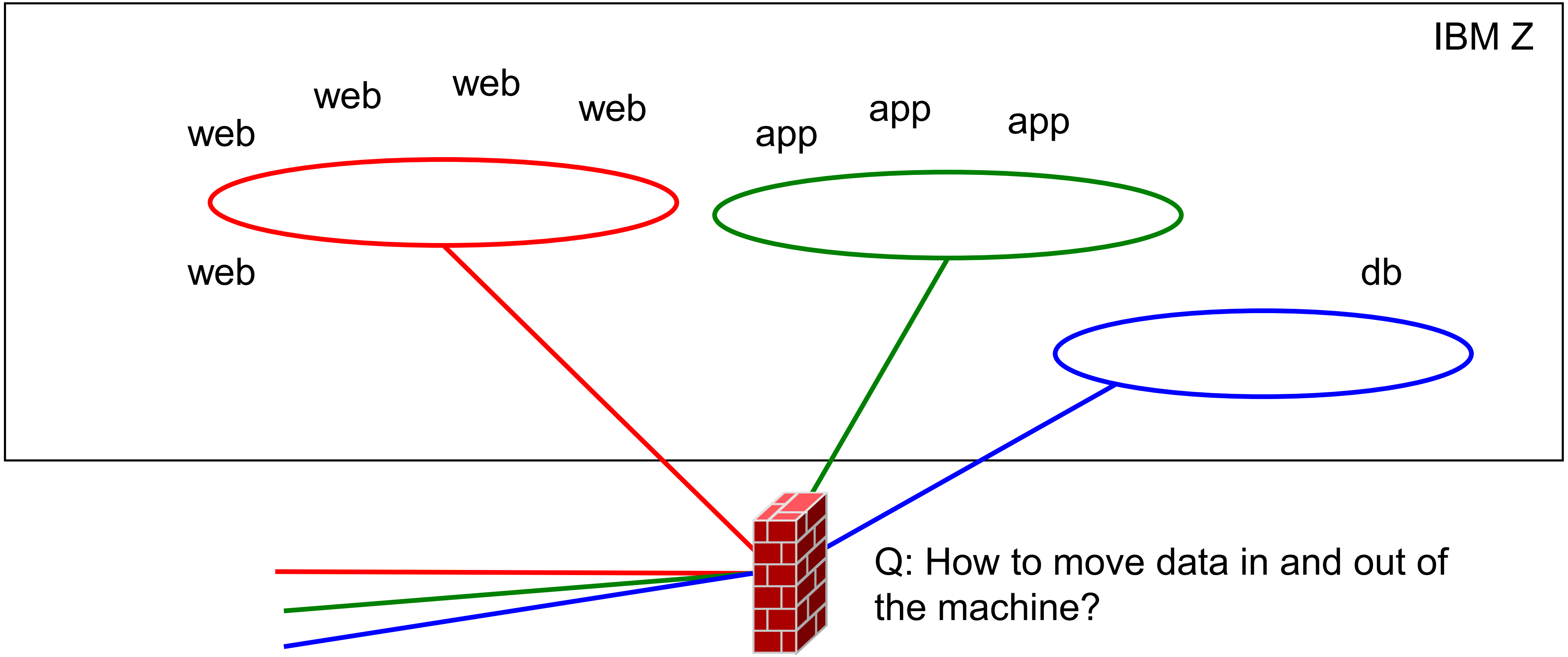
Notes

References to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe on any of the intellectual property rights of IBM may be used instead. The evaluation and verification of operation in conjunction with other products, except those expressly designed by IBM, are the responsibility of the user.

Topics

- Overview
- The Uplink
- The virtual NIC
- The VSWITCH controller
- Link Aggregation
- Sharing OSAs
  - Global VSWITCH and shared port groups
- HiperSocket VSWITCH Bridge
- Diagnostics

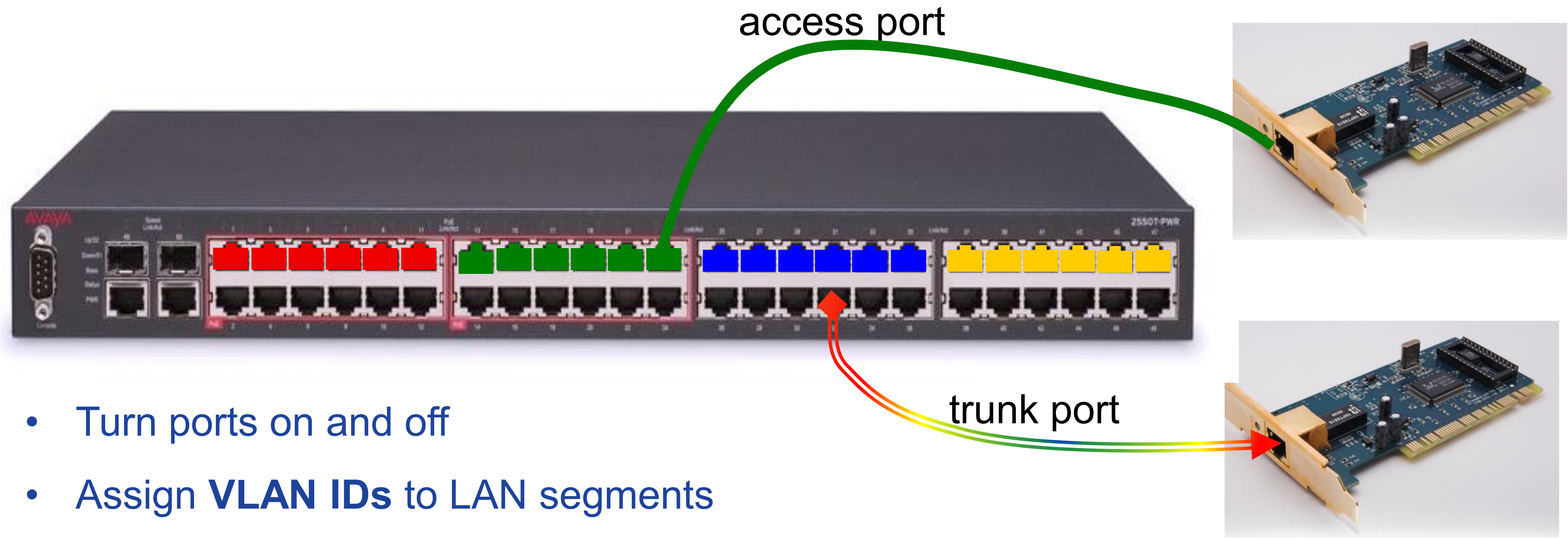
Multi-zone Network on IBM zSystems
With outboard firewall / router



Q: How to move data in and out of the machine?

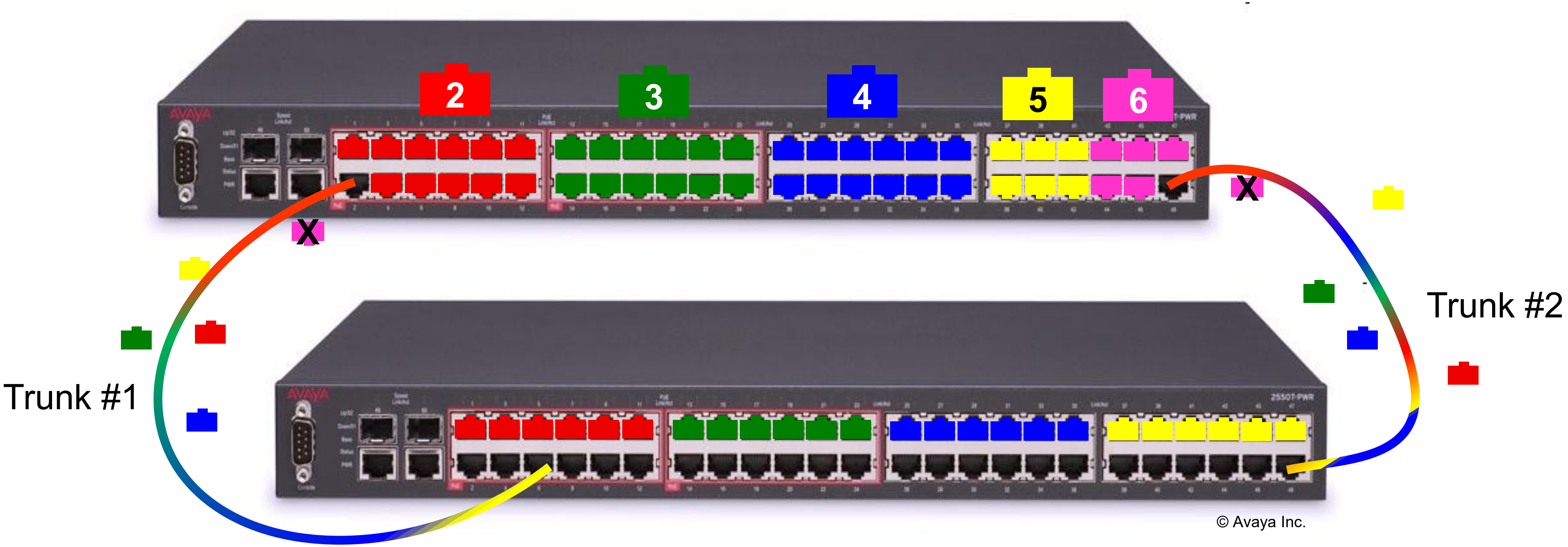
A: z/VM® Virtual Switch

Q: What’s a switch?
A: A network device management endpoint



- Turn ports on and off
- Assign **VLAN IDs** to LAN segments
- Associate **access** ports with a single VLAN ID
- Associate **trunk** ports with multiple VLAN IDs
- Provide fast switching of data between ports
- Provide sniffer functions

Q. What’s a Bridge?
A: A way to connect two switches



© Avaya Inc.

- If you run out of ports, you don’t throw it away, you **bridge** it to an adjacent switch
- A **trunk** port carries ethernet frames for **multiple** LAN segments (subnets)
- **VLAN tags** in each frame identify the LAN segment it belongs to
- Redundant connections for high availability

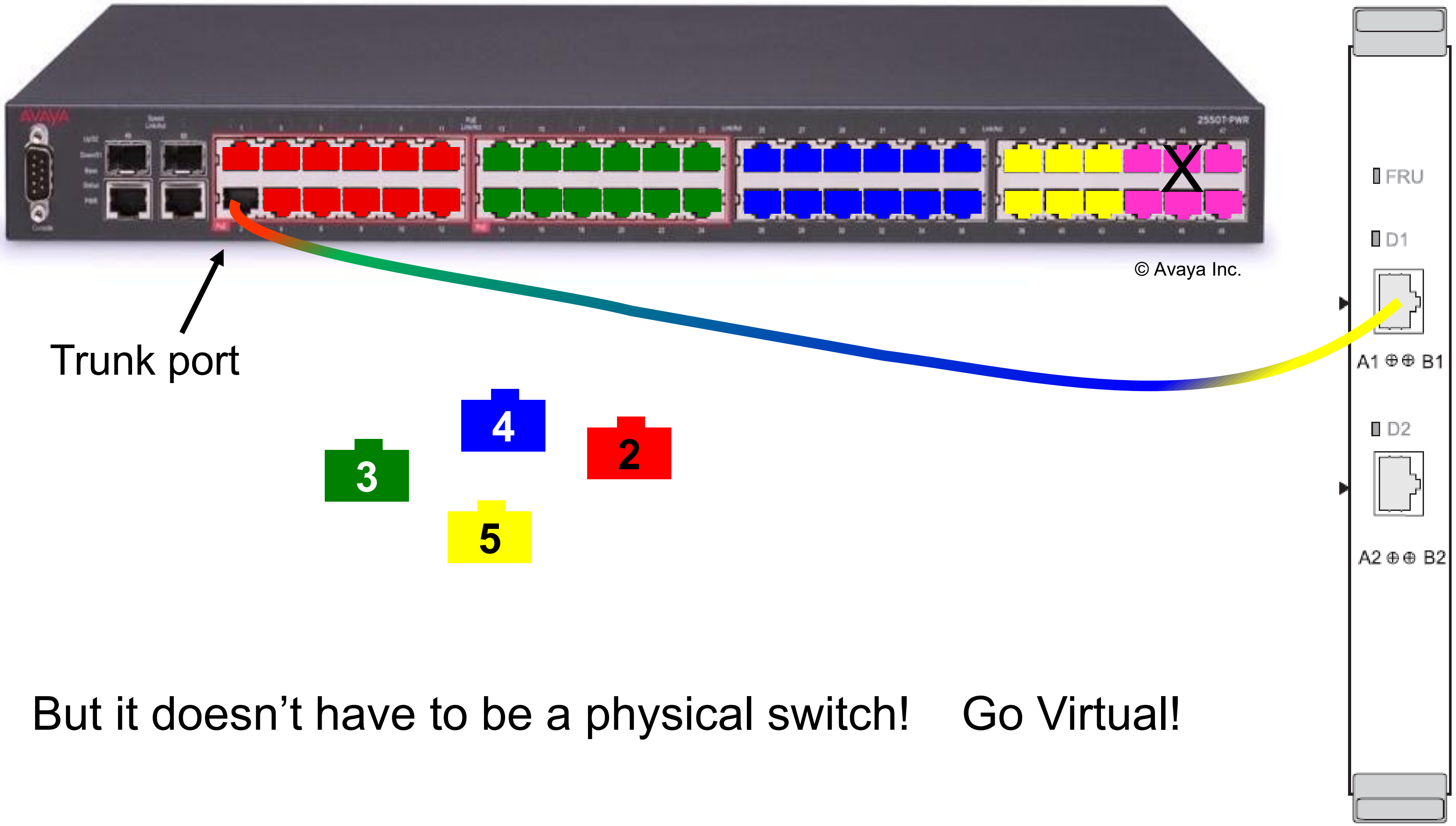
Bridge versus Router

- A bridge connects two LAN segments that are in the same subnet
  - aka "Layer 2 switch"
  - Behaves as a single LAN segment
  - Do not confuse this with deprecated term "Layer 2 VSWITCH"

- A router connects two LAN segments that are in different subnets
  - aka "Layer 3 switch"
  - Do not confuse this with deprecated term "Layer 3 VSWITCH"

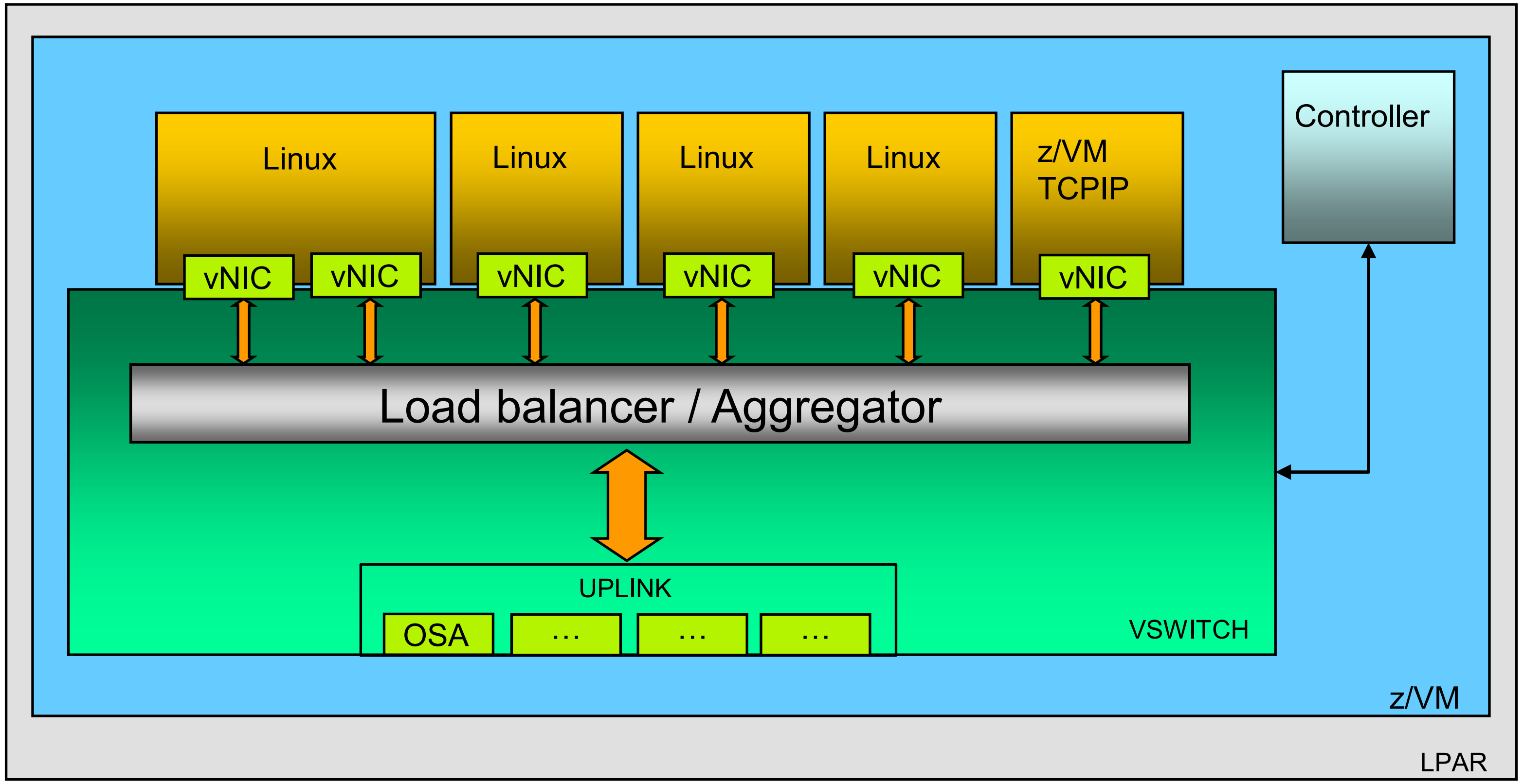
- A VSWITCH configurations are **bridges,** not routers.

VLAN-aware Virtual Switch



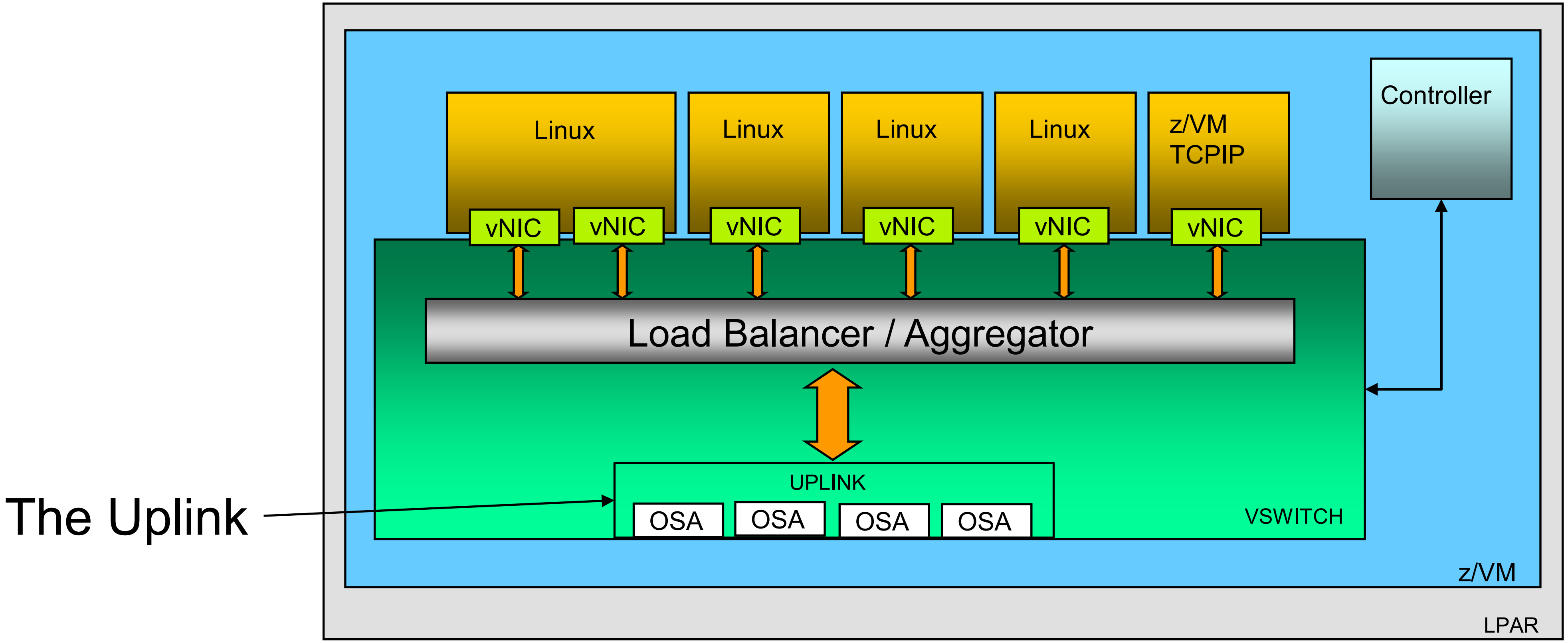
But it doesn’t have to be a physical switch!   Go Virtual!

The Virtual Switch



The Virtual Switch

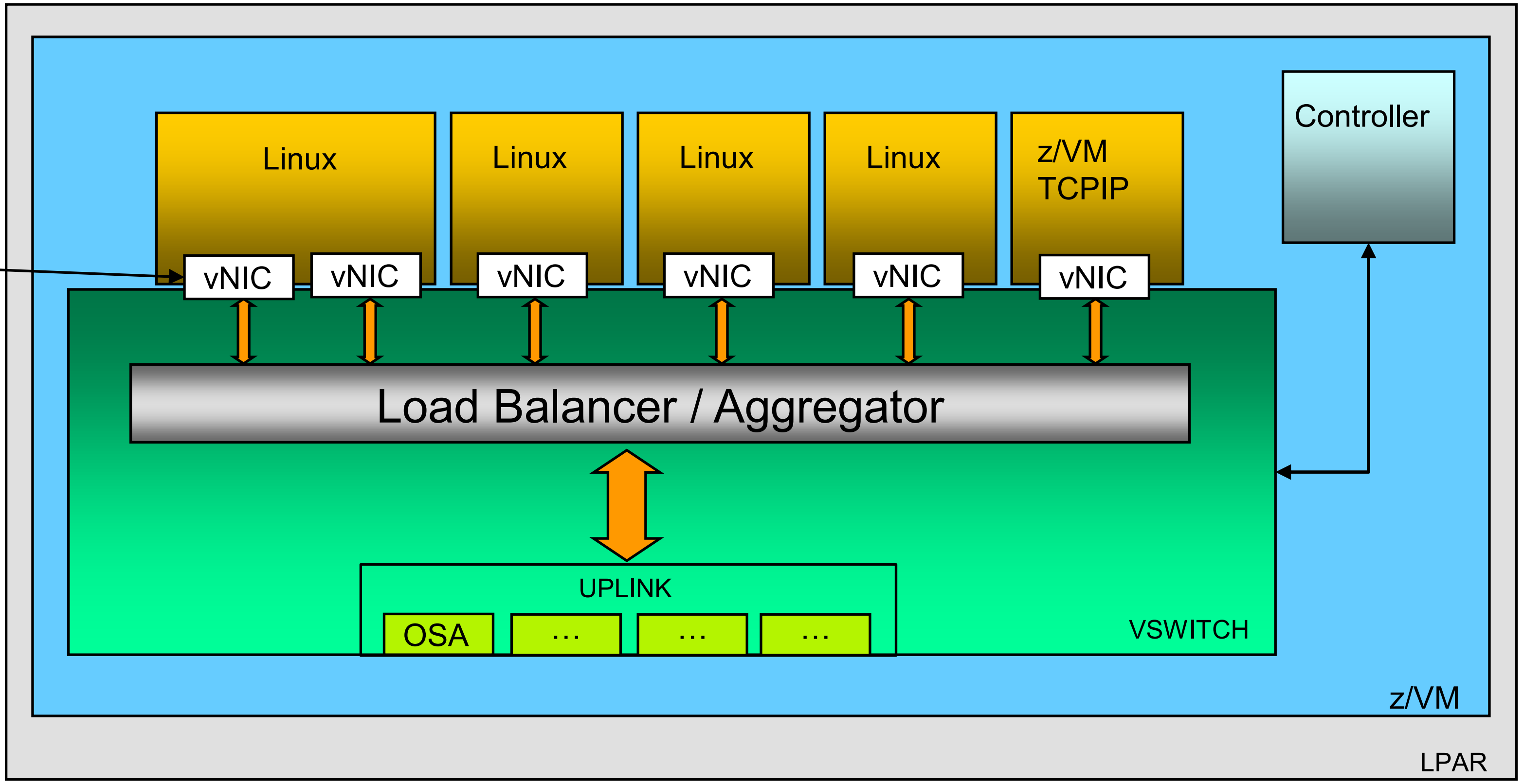
Configurable Elements



The Virtual Switch

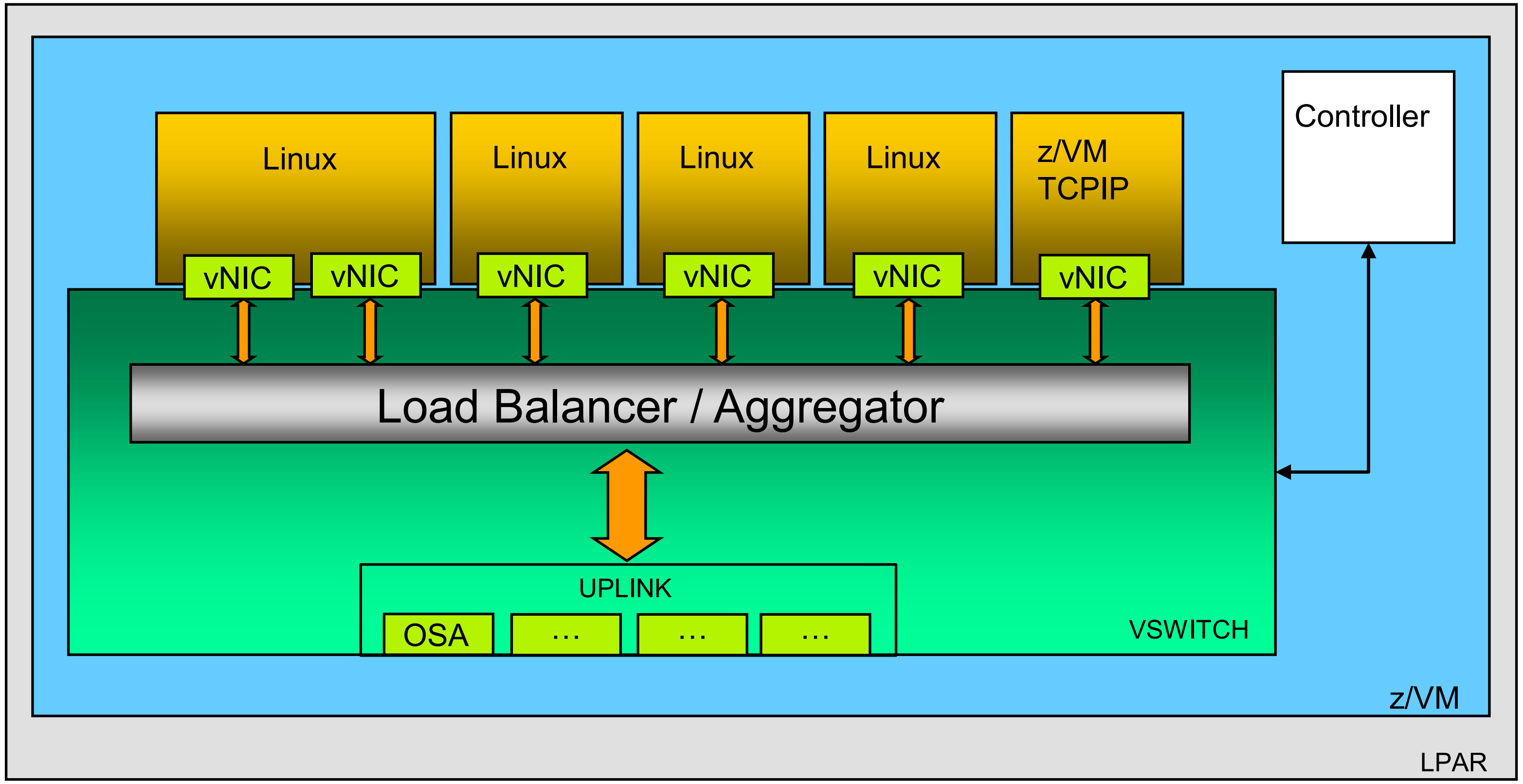
Configurable Elements

The virtual network interfaces



The Virtual Switch

Configurable Elements



The controller

Virtual Switch general principles

- Bring controllers up first

-
```
DEFINE VSWITCH name {ETHERNET | IP} PORTBASED

             {uplink attributes}
             {virtual NIC defaults}
             {accounting settings}
```

Suggested Practice:

- Use PORTBASED option for consistency of QUERY VSWITCH output and future directions
- Bring up in AUTOLOG1 / 2 after the controllers are up

- Unless otherwise configured, traffic remains as close to the virtual machines as possible
  - Within the VSWITCH
  - Within the OSA
  - Out to the physical switch

The Uplink

Uplink Port

- Connects VSWITCH to network
  - Without an uplink, data can move only among coupled guests
    - Better than a Guest LAN!

- Operates in ETHERNET or IP mode

- VLAN aware or unaware

- For high availability, you need more than one physical connection
  - Single-port failover
  - Link Aggregation port group

Uplink: IP mode

-
  ```
  DEFINE VSWITCH name   IP  PORTBASED
               [NONROUTER | PRIROUTER]
  ```

- Guest and host device driver sends and receives IP packets

- All virtual NICs have the same physical MAC address
  - Packet delivery based on guest-registered IP address

- Good for z/OS guests (they can’t use ETHERNET mode), but
  - No IPv6
  - No DHCP
  - No link aggregation

Suggested Practice:

- Use IP mode only for z/OS guests

Uplink: ETHERNET mode

- `DEFINE VSWITCH *name* ETHERNET PORTBASED`

- Guest and host device driver sends and receives fully-formed ethernet frames

- Virtual MAC address used as physical MAC address
  - More about that later

- No z/OS

Uplink: OSA port options

No ports

- Similar to Guest LAN, but with better security
- Excellent for 2nd level systems

One active port with one or two failover ports

- Round-robin failover
- If all dead, wait for signs of life
- SET VSWITCH SWITCHOVER to manually change
- Maximum bandwidth = 25 Gb/s

Up to 8 active ports operating concurrently

- IEEE 802.1AX link aggregation (a form of channel bonding)
- Maximum bandwidth = 200 Gb/s (8 x 25 Gb/s)
- SET PORT GROUP to add or remove ports
- ETHERNET mode only

Uplink: OSA port selection

```
DEFINE VSWITCH …

             RDEV NONE
or
             RDEV port1  [port2  [port3] ]
or
             GROUP group_name
```

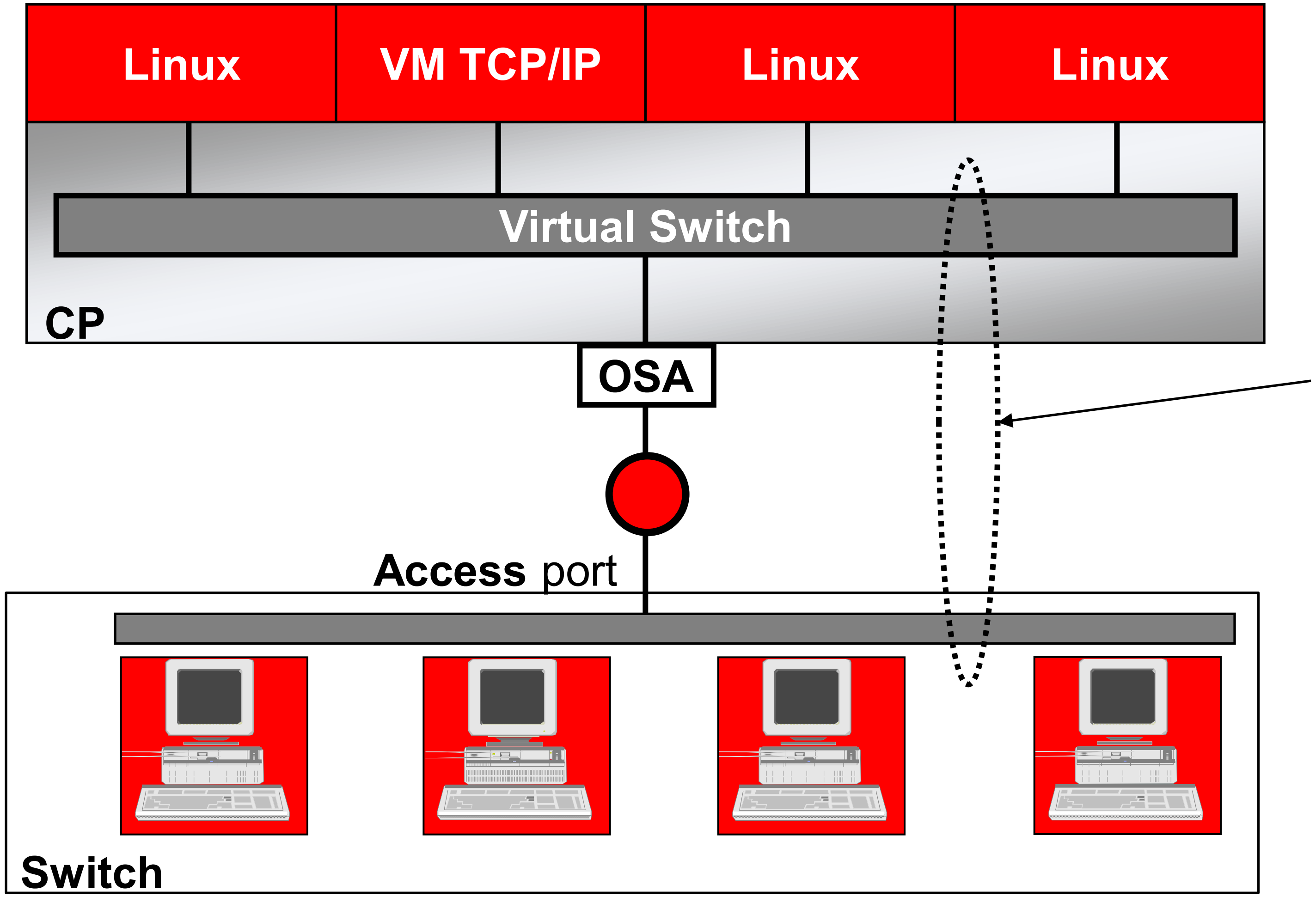
- RDEV NONE is a ***disconnected*** VSWITCH

- Port is identified by device number (points to an OSA PCHID) and an optional physical port specification (P0 or P1)
  - 1EC0   (default is P0)
  - 1EC0.P0
  - 1EC0.P1

- Group name comes from **SET PORT GROUP**

Uplink: Access port, VLAN unaware

VSWITCH carries traffic for a single LAN segment (subnet)



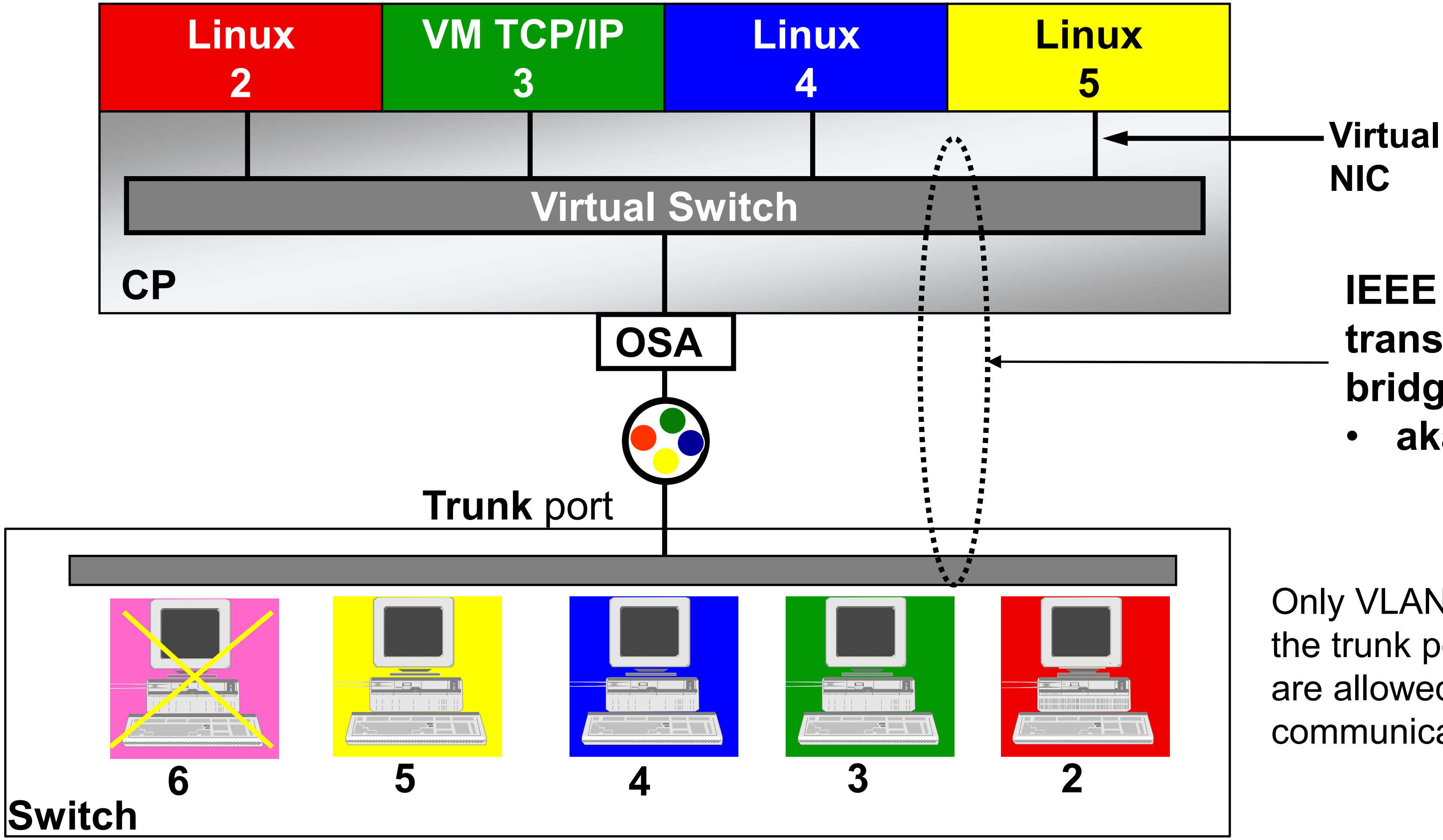
Transparent Bridge

- Same LAN segment
- No 802.1q

Only the **VLAN id** configured for the access port can be used

Uplink: Trunk port, VLAN aware

VSWITCH carries traffic for multiple LAN segments (subnets)

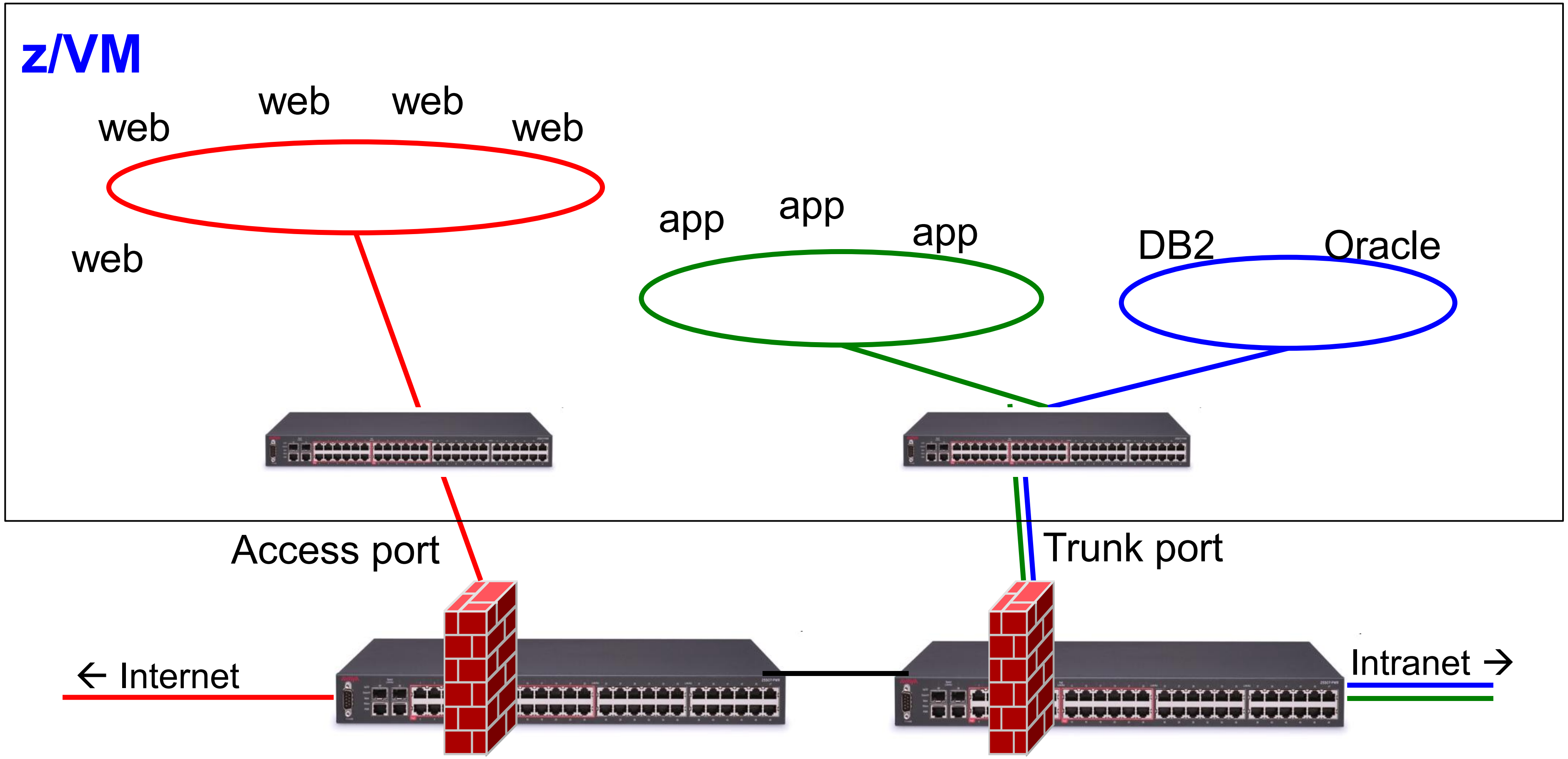


IEEE 802.1Q transparent bridge

- **aka “dot1q”**

Only VLANs configured for the trunk port by the switch are allowed to communicate

VLAN-aware Virtual Switch
Multiple LAN segments per VSWITCH



Shared infrastructure is cheaper than dedicated, but be aware of any rules that prohibit comingling of Internet and Intranet traffic on the same infrastructure

Uplink: Trunk or Access port?

— **Access** port

```
DEFINE VSWITCH …
            VLAN UNAWARE
```

- This is the default configuration

— **Trunk** port

```
DEFINE VSWITCH …
            VLAN AWARE | vid
            NATIVE 1 | NATIVE vid | NATIVE NONE
```

Suggested Practices

- Use a trunk port defined using “VLAN AWARE NATIVE NONE”
- Don’t specify PORTTYPE TRUNK (it doesn’t do what you think it does) ← ← ←

Sidebar: A word of advice about the native VLAN

- When an untagged frame is received on a trunk port the switch will associate the frame with the local default or native VLAN ID (VID), typically VLAN 1

- Used for switch management traffic
  - **Do not allow guests to interfere with the physical switch!**

- Identified by the NATIVE keyword on the DEFINE VSWITCH command
  - CP removes tags for frames associated with the native VLAN ID

- **VLAN nn NATIVE nn** is **wrong!**
  - Same number on both operands
  - You're really plugged into an **access port**
  - Change to VLAN UNAWARE
  - If any NICDEF has an assigned VLAN id that matches NATIVE nn, it's wrong, too!

Disable class G user's ability to create a Guest LAN

- VMLAN statement in SYSTEM CONFIG:

```
VMLAN
  LIMIT TRANSIENT 0
```

Suggested Practice

- Don’t use Guest LANs – use disconnected VSWITCH instead

The Virtual NIC

Virtual NIC

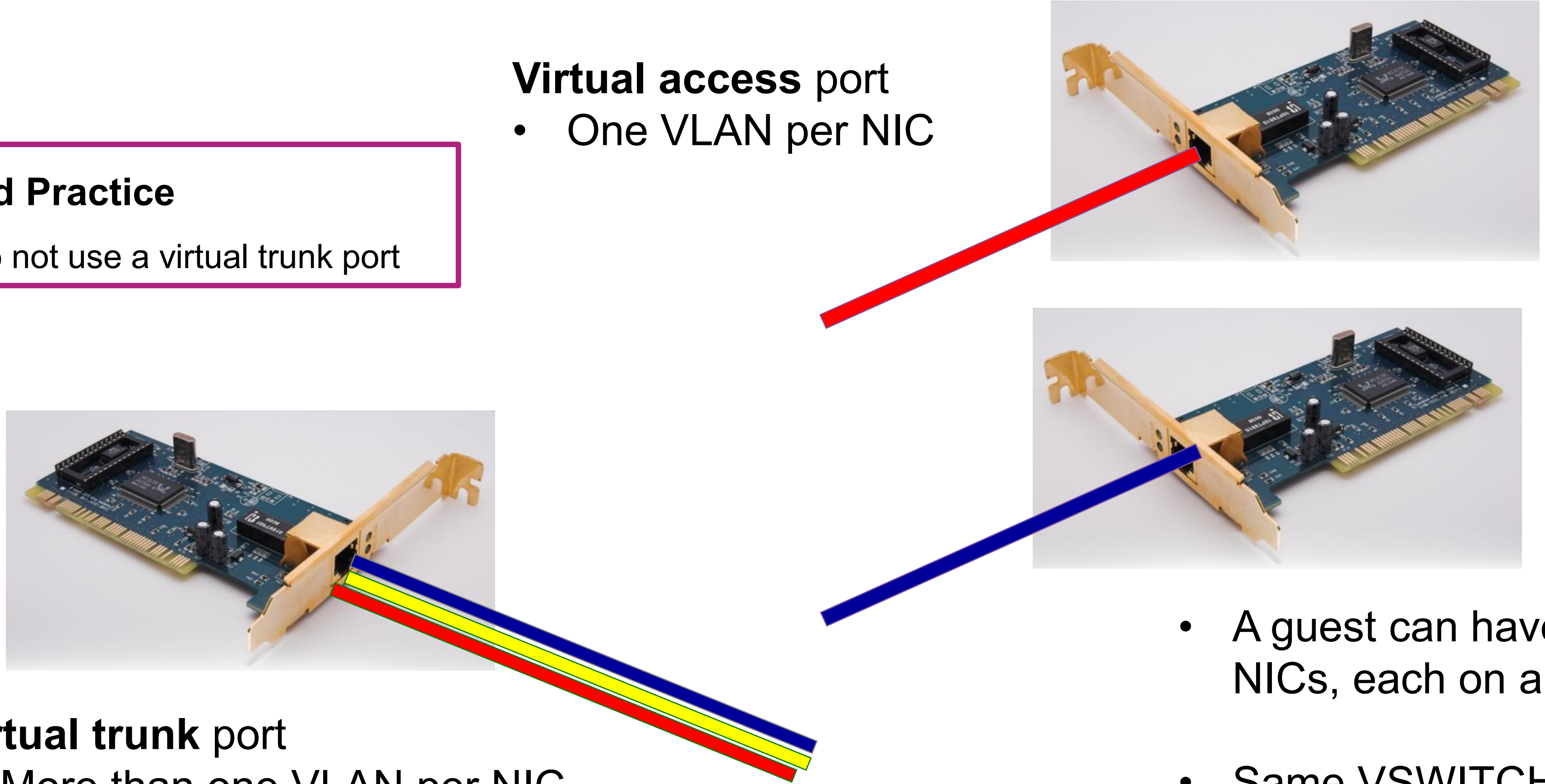
Suggested Practice

- Do not use a virtual trunk port

Virtual access port

- One VLAN per NIC

✓



☒

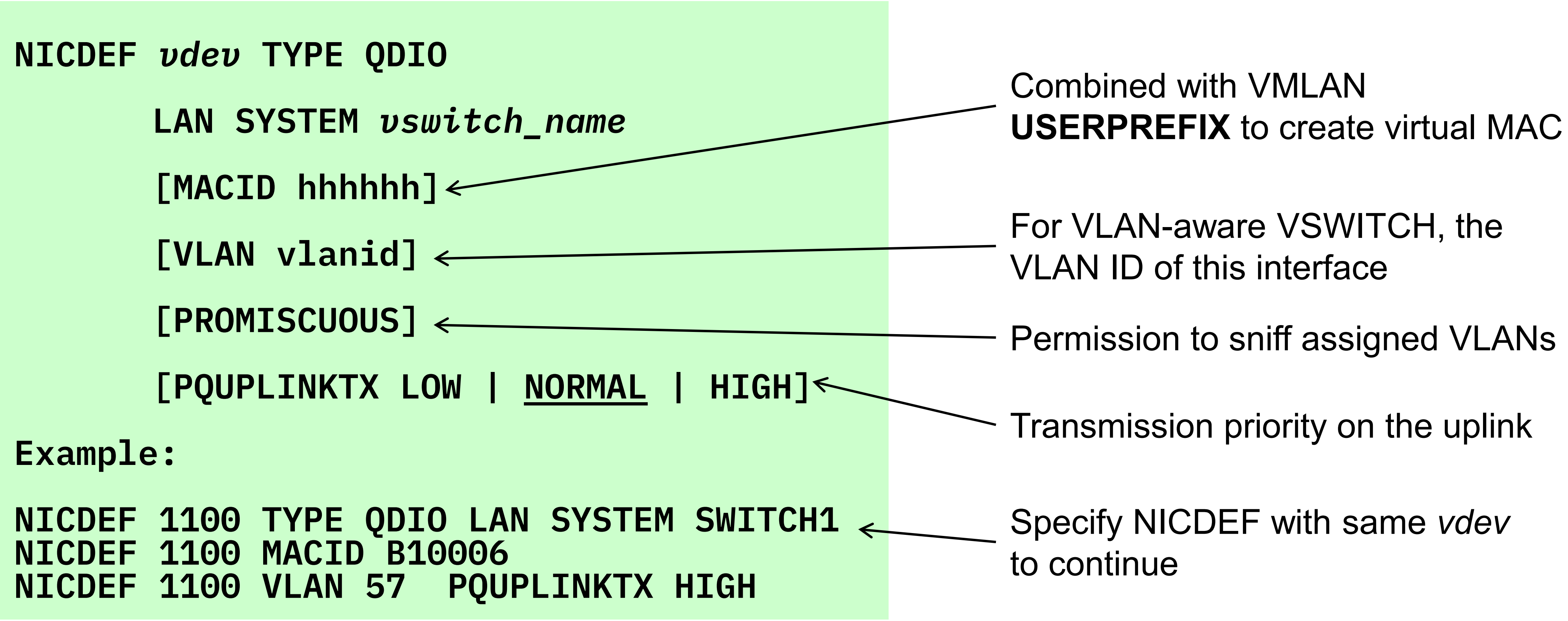
Virtual trunk port

- More than one VLAN per NIC
- Requires more processing by the guest

- A guest can have multiple virtual NICs, each on a different VLAN
- Same VSWITCH with different VLANs
- Different VSWITCH

Virtual NIC: User Directory

Interface is fully configured in the user's directory entry



```
NICDEF vdev TYPE QDIO
       LAN SYSTEM vswitch_name
       [MACID hhhhhh]
       [VLAN vlanid]
       [PROMISCUOUS]
       [PQUPLINKTX LOW | NORMAL | HIGH]
Example:

NICDEF 1100 TYPE QDIO LAN SYSTEM SWITCH1
NICDEF 1100 MACID B10006
NICDEF 1100 VLAN 57  PQUPLINKTX HIGH
```

Combined with VMLAN **USERPREFIX** to create virtual MAC

For VLAN-aware VSWITCH, the VLAN ID of this interface

Permission to sniff assigned VLANs

Transmission priority on the uplink

Specify NICDEF with same *vdev* to continue

Automatically creates *vdev, vdev*+1, and *vdev*+2

Virtual NIC: MAC Addresses

- 6 bytes
  - E.g. 02:00:0A:00:01:23
  - Prefix + ID

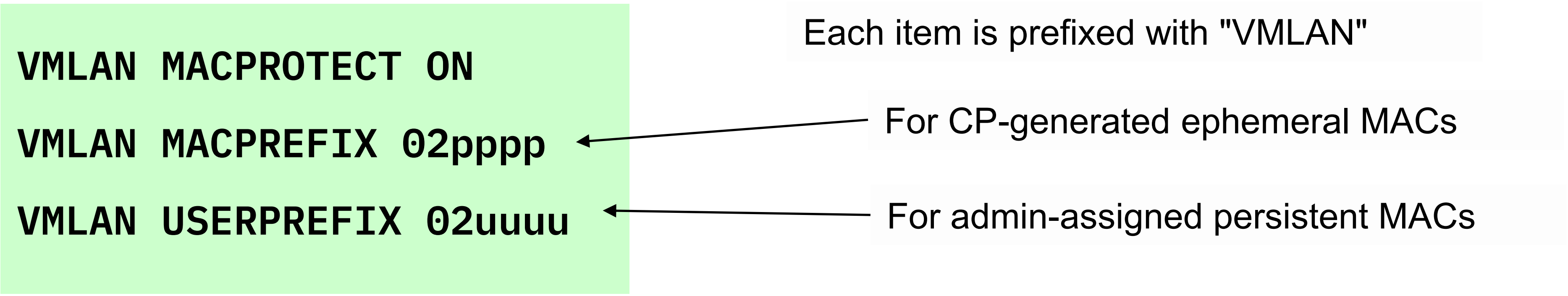
- Prefix
  - E.g. 02:00:0A
  - Comes from **VMLAN** statement in SYSTEM CONFIG
    - Leading ‘02’ is required; indicates that they are administratively-defined addresses, not globally unique

- ID
  - E.g. 00:01:23
  - Persistent: From **MACID** operand of NICDEF directory entry
  - Ephemeral: If not defined, set by CP

- MAC will appear on the physical network
  - ETHERNET mode VSWITCH only

Virtual NIC: Controlling the MAC address

— Global attributes in the **VMLAN** statement in SYSTEM CONFIG:



```
VMLAN MACPROTECT ON

VMLAN MACPREFIX 02pppp

VMLAN USERPREFIX 02uuuu
```

Each item is prefixed with "VMLAN"

For CP-generated ephemeral MACs

For admin-assigned persistent MACs

Suggested Practices

- MACPROTECT ON prevents guests from changing their assigned MAC address
- MACPREFIX unique per z/VM instance
  - Do not allow to default to 020000 (that’s how you can detect a misconfigured system!)
  - Enforced for SSI
- USERPREFIX same across all members of a shared directory cluster
  - Enforced for SSI

Virtual NIC: Sniffers

— **Promiscuous** mode for sniffers

- Guest must be authorized via NICDEF

- Guest enables promiscuous mode using CP SET NIC or via device driver controls
    - E.g. tcpdump –P and download for Wireshark

- Guest receives copies of all frames sent or received for all authorized VLANs

Virtual NIC: Priority Queuing

- OSA Express **Priority Queuing** enables the host to provide an ordered set of outbound data queues that OSA will service in order, but without queue starvation.

- CP creates four queues (in priority order):
  - System
  - High (guest)
  - Normal (guest)
  - Low (guest)

- You assign priority to each virtual NIC
  - Default is “normal”

- Activation required

```
DEFINE VSWITCH …
             PRIQUEUING ON
```

The Controller

VSWITCH Controller

- Virtual machine that handles OSA housekeeping duties
  - Specialized VM TCP/IP stack to start, stop, monitor, and query OSA
  - Each controller can service any number of VSWITCHes
  - **Not involved in data transfer**

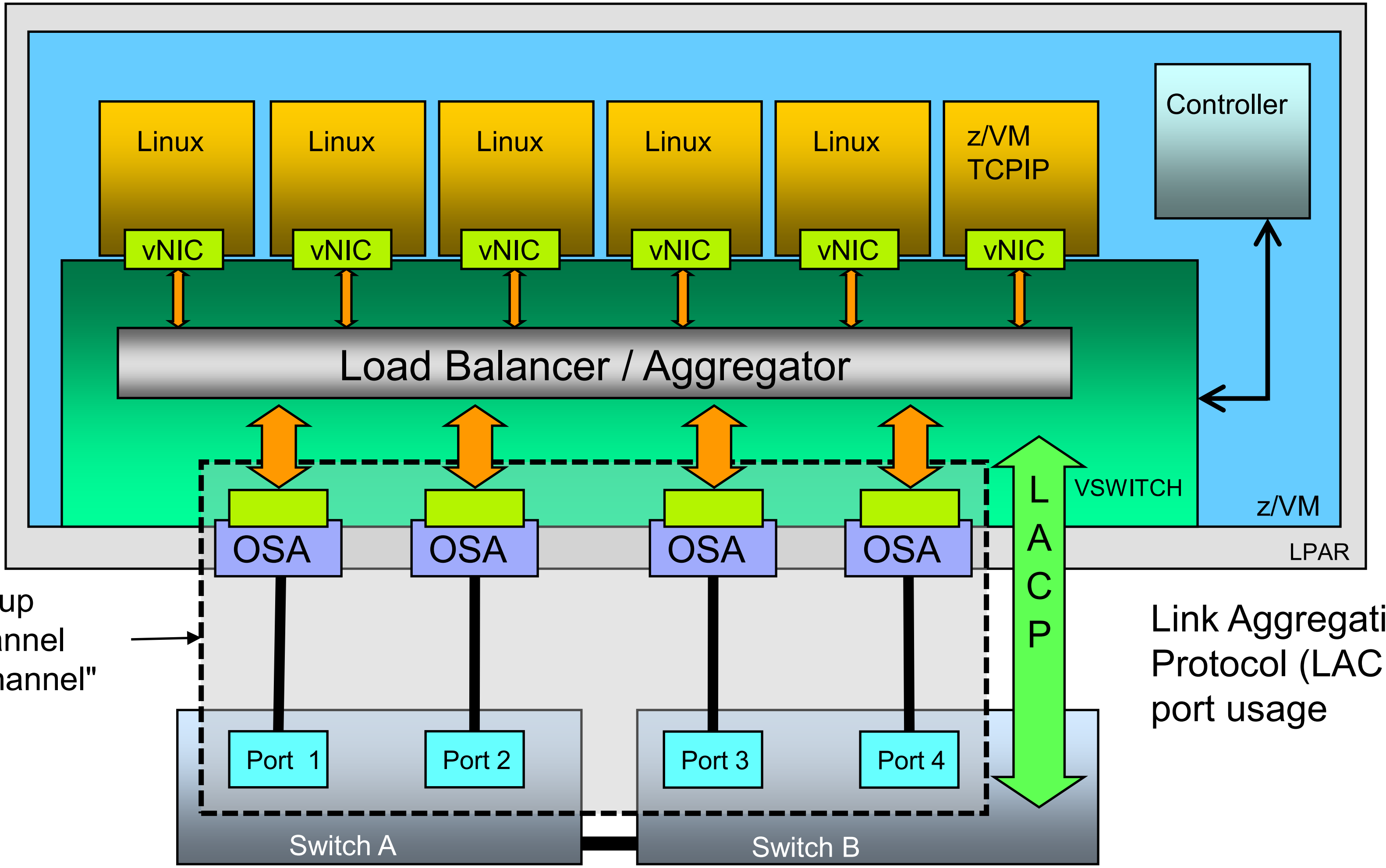
- DTCVSW1-DTCVSW4
  - Except for obey list, do not modify their configurations unless directed by Support Center
  - Monitor with system automation and keep them logged on
  - Automatic failover
  - If no controllers are available, uplink will stop!
    - Guest-guest communication ok

- Issues messages to virtual console during error recovery
  - `NETSTAT CP CLOSE CONS TO` *`userid`* `(TCP DTCVSW`*`x`*

IEEE 802.1AX
Link Aggregation

Link Aggregation

Non-disruptive scalability and failover



- Port group
- Port channel
- “Etherchannel"

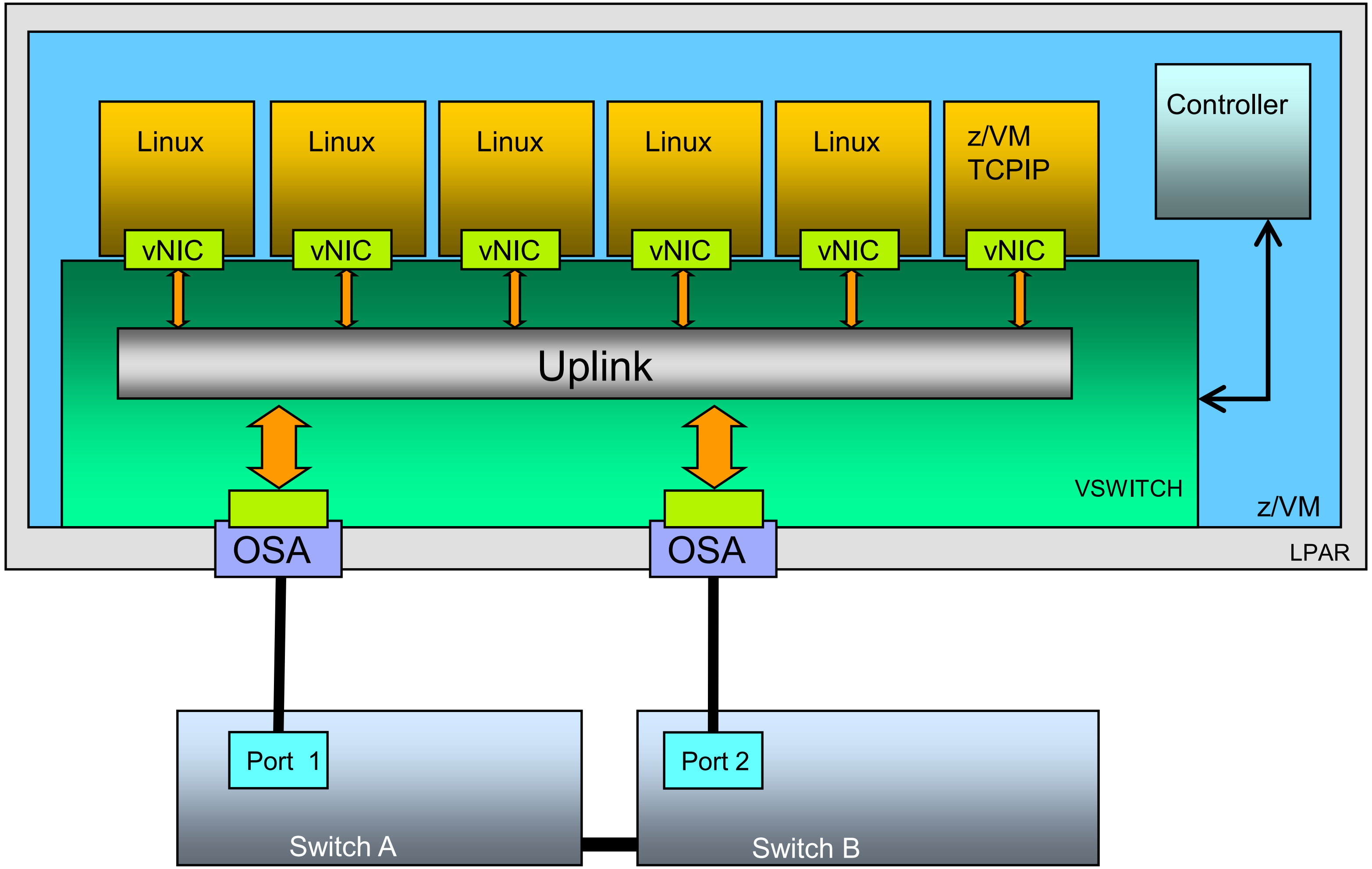
Virtual NIC assigned to a real NIC

VSWITCH or switch can initiate a change in the assignment via LACP.

Link Aggregation Control Protocol (LACP) controls port usage

Virtual Chassis VC VSS “stacked” switches

Failover (non-Link Aggregation)



Virtual Chassis VC VSS “stacked” switches

Link Aggregation

- Binds multiple OSA-Express ports into a single pipe
  - Up to 8 OSA ports per virtual switch
  - Increases Virtual Switch bandwidth
  - Provides seamless failover in the event of a failed OSA, switch port, cable, or switch
  - Only supported for ETHERNET VSWITCHes
  - Virtual NIC is still limited to bandwidth of single OSA
  - Also called a **port channel** or **Etherchannel**

- With **virtual chassis** or **stacked switch** support from switch vendor, can also handle physical switch outage

- Switches talk to each other to provide load balancing and to add/remove adapters from port group

Link Aggregation: Port group

- Create an OSA port group

```
SET PORT GROUP PCHNL01 JOIN F100 F200.P1
```

- Create a VSWITCH that references to group

```
DEFINE VSWITCH …
             ETHERNET
             GROUP PCHNL01
```

- Done and dusted!
- OSA ports cannot be shared with other VSWITCHes or LPARs unless using **shared port groups**

Suggested Practices

- Name your port groups to match the name of the port channel on the switch
- Put VSWITCH definition in AUTOLOG1 or AUTOLOG2

Sharing OSAs

Sharing OSAs without Link Aggregation

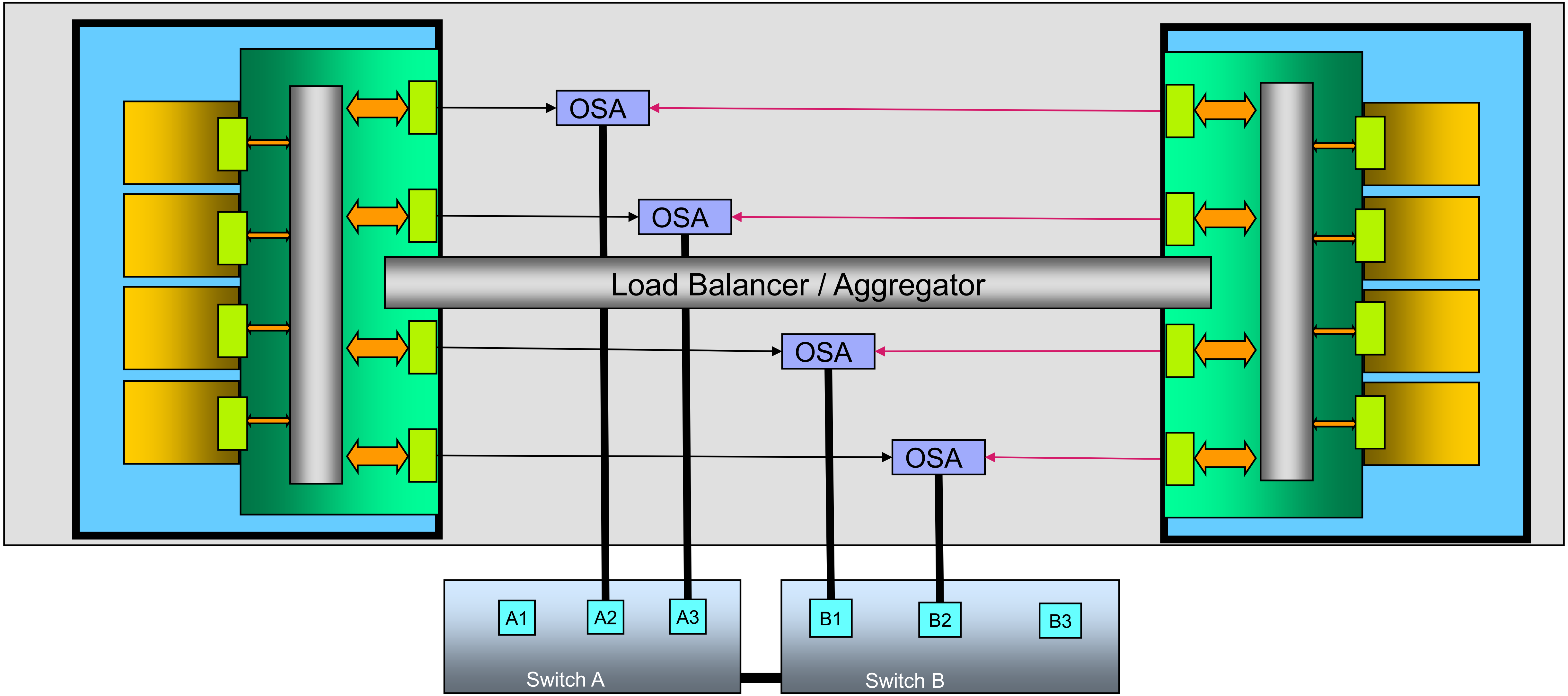
- No special restrictions
- All operating systems
- VSWITCH and/or dedicated

Sharing OSAs with link aggregation
Why?

- If suggested practice is 4 OSA ports per VSWITCH
  - … and you have a 4-member SSI cluster
  - … and you have one VSWITCH per member
  - … and you cannot share OSAs that are in a link-aggregation port group
  - … then you need **16** ports (i.e. 16 10Gb OSA-Express features)

- That’s **¼** of the OSA capacity of the machine (and expensive)!

Sharing OSAs with Link Aggregation
Global VSWITCH with shared port group



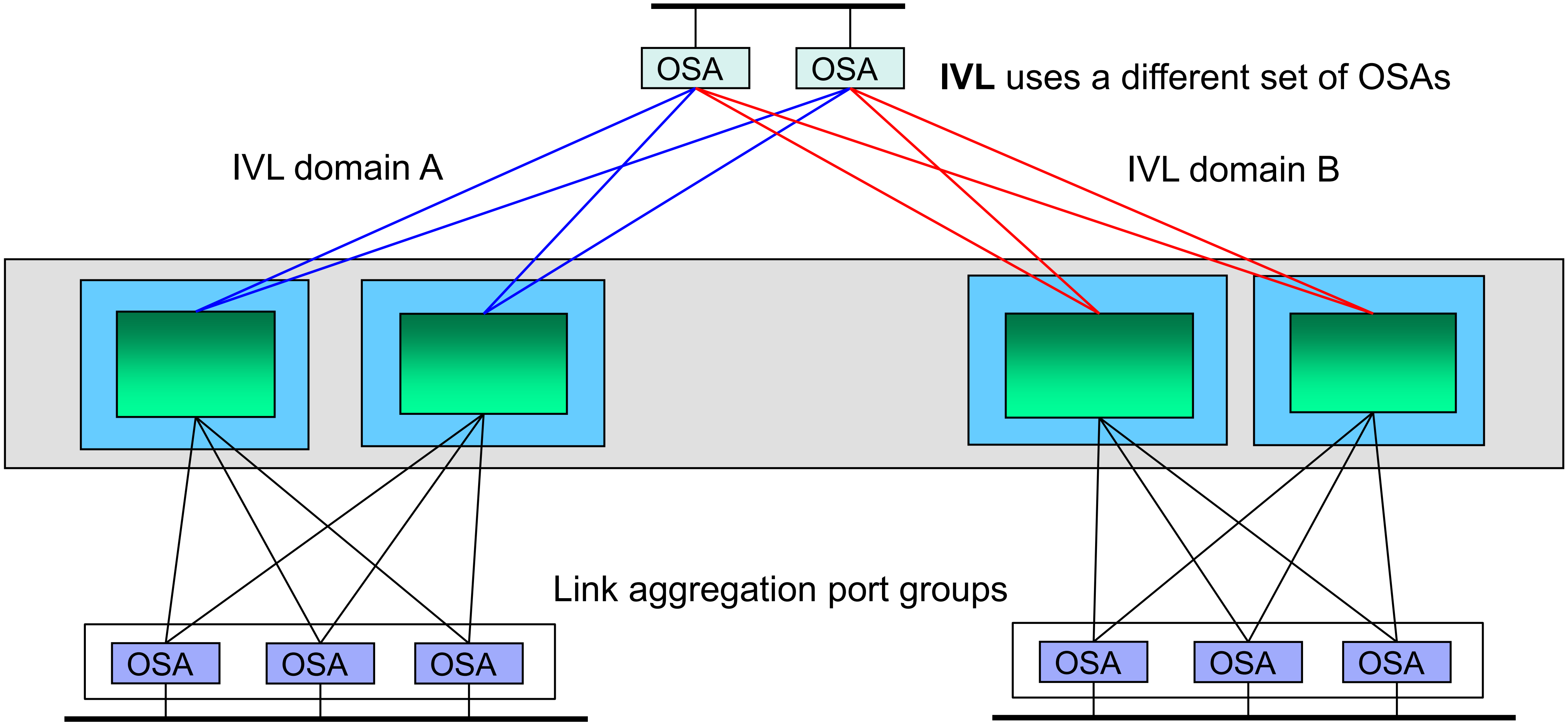
Shared Link Aggregation Port Groups

- Two new system constructs
  - **Global VSWITCH**
    - Virtual Switch that spans multiple z/VM LPARs within a single CPC, all using the same link aggregation port group
  - **Inter-VSWITCH Link** (IVL)
    - Provides data channel for management of shared port groups and the Global VSWITCH
    - Each z/VM system is assigned to one **IVL domain** (A – H)
    - Up to 16 systems in a domain
- All members of the domain can use a SHARED port
  - If not shared, the early bird catches the worm!
- Configuration changes to shared port group or global VSWITCH are propagated to all members of the domain

Suggested Practice

- One domain for production, another domain for dev/test

Shared Link Aggregation Port Groups



IVL uses a different set of OSAs

IVL domain A

IVL domain B

Link aggregation port groups

IVL: Create the IVL VSWITCH

```
DEFINE VSWITCH name TYPE IVL  DOMAIN d  [VLAN vid]
```

— Conventional RDEV list or exclusive port GROUP

- Remember to provide OSA port redundancy!
- **No, the IVL cannot use the same OSAs that the global VSWITCHes are using as uplinks!**

— Do this on each z/VM that will share the port group

- Command must be the same on all instances (name, domain, VLAN id)
- QUERY VSWITCH will show the name as *systemid.name* instead of “SYSTEM *name*”
  - If you have any programs that interpret the output of QUERY VSWITCH, you may need to fix them

— z/VM automatically joins the domain

— Do this before you create a shared port group or global VSWITCH

IVL: Dynamic Controls

```
SET VSWITCH name IVLPORT option
```

Options:

- VLAN - Change the VLAN ID associated with the IVL
- RESET - Terminate and recreate the IVL port connection
- PING - Tests connectivity between z/VM hypervisors in the same IVL domain
  - **`set vswitch`** ***`name`*** **`ivlport ping all`**
- HEARTBEAT TIMEOUT - Adjusts how often the local z/VM system confirms connectivity with the other domain members

Create a shared Port Group

```
SET PORT GROUP name LACP ACTIVE SHARED
SET PORT GROUP name JOIN rdev1.port rdev2.port …
```

- Device numbers can be any device number on the chpid
- CP will select the device numbers to be used on the other z/VM instances
- CP propagates changes to the port group configuration to all active members of the IVL domain
- Do this before you create a global VSWITCH

Create a Global VSWITCH

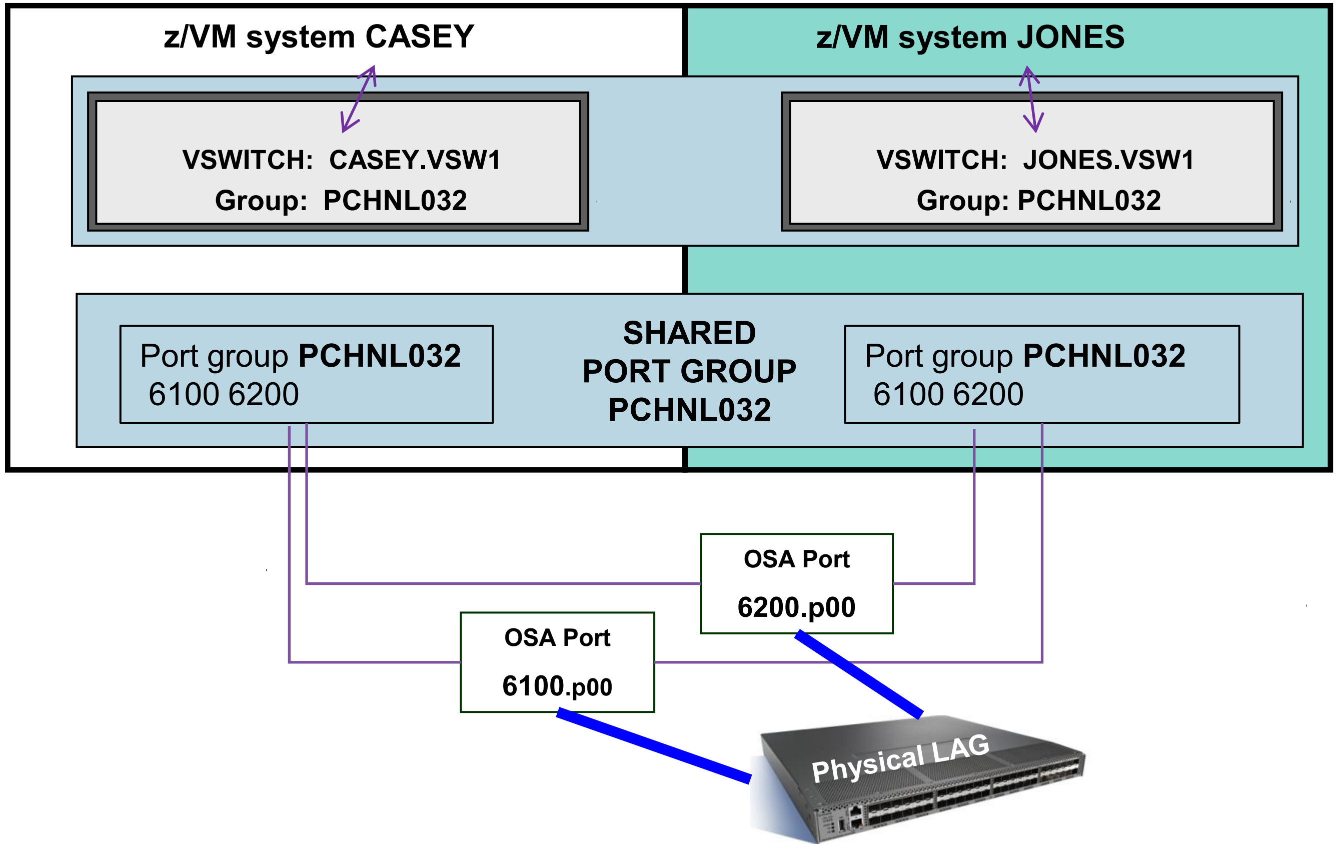
```
DEFINE VSWITCH name GLOBAL ETHERNET GROUP group
```

- Multiple global VSWITCHes can be defined per z/VM instance
  - All in the same IVL domain

- An *instance* of a Shared Port Group is created when it is configured to a virtual switch

Create a Global VSWITCH: Example

```
SET PORT GROUP PCHNL032 LACP ACTIVE SHARED
SET PORT GROUP PCHNL032 JOIN 6100 6200
DEFINE VSWITCH VSW1 GLOBAL ETHERNET GROUP PCHNL032
```



- Up to 4 global VSWITCHes in the same **partition** can share a port group
- A 2nd level VSWITCH counts!

Link Aggregation: Asynchronous Port Group and VSWITCH Initialization !ALERT!

- Guests cannot connect to a VSWITCH until it is defined (virtual NIC errors)
- A VSWITCH using a shared port group will not be defined until the port group is ready
- Shared port group cannot form until physical switch and VSWITCHes reach agreement
- The SET PORT GROUP and DEFINE VSWITCH commands will complete asynchronously

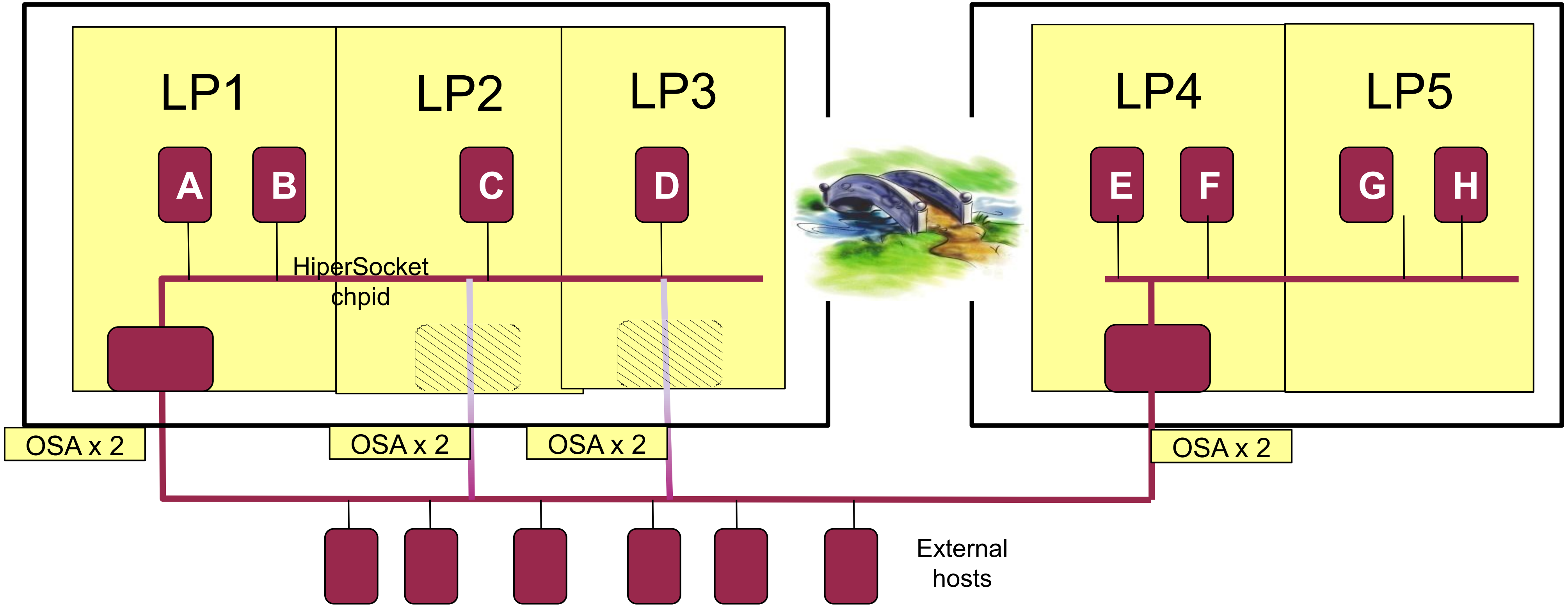
- **Placing SET PORT GROUP and DEFINE VSWITCH in SYSTEM CONFIG is not sufficient!**

- If you bring guests up before your VSWITCH is defined, guests will get NIC errors

- Defer guest startup to automation (e.g. IBM Operations Manager) which waits for VSWITCH activation
  - Watch for messages to OPERATOR
  - QUERY-style polling logic

HiperSocket VSWITCH Bridge

HiperSocket VSWITCH Bridge



- One active bridge per partition

- Path MTU discovery support
  - Large frames inside
  - Smaller frames outside

HiperSocket VSWITCH Bridge

- Connect HiperSocket LAN to ethernet LAN without a router
  - Same subnet as ethernet LAN

- Full redundancy
  - Up to 5 bridges per CPC (CEC)
  - Automatic failover with optional failback
  - Each bridge can have more than one OSA uplink (typical)

- Enables cross-CPC Live Guest Relocation for guests using HiperSockets

- Does not work with z/OS LPARs
  - Look at z/OS HSCI

HiperSocket VSWITCH Bridge

```
DEFINE VSWITCH …
             ETHERNET
             BRIDGEPORT RDEV hs-rdev
             [PRIMARY]
```

- I/O configuration change required
  - HiperSocket CHPID must be defined with CHPARM=x4

- The EXTERNAL_BRIDGED operand is available on CP DEFINE CHPID command if using native z/VM dynamic I/O

Suggested Practices

1. Use **ETHERNET** mode VSWITCH with link aggregation
2. Do not specify other options on DEFINE VSWITCH unless you study them carefully
   - E.g. PORTTYPE TRUNK (boo! hiss!)
3. Specify **MACPROTECT ON** and **LIMIT TRANSIENT 0** on VMLAN statement in SYSTEM CONFIG
4. VLAN-aware VSWITCH should be defined with **VLAN AWARE NATIVE NONE**
5. Don’t use virtual trunk ports – leave guests VLAN-**unaware**
6. Put VSWITCH definitions in AUTOLOG1 or AUTOLOG2
   - Don't use SYSTEM CONFIG – causes stacking on DTCVSW1

Diagnostics

- CP QUERY VMLAN
  - to get global VM LAN information (e.g. limits)
  - to find out what service has been applied

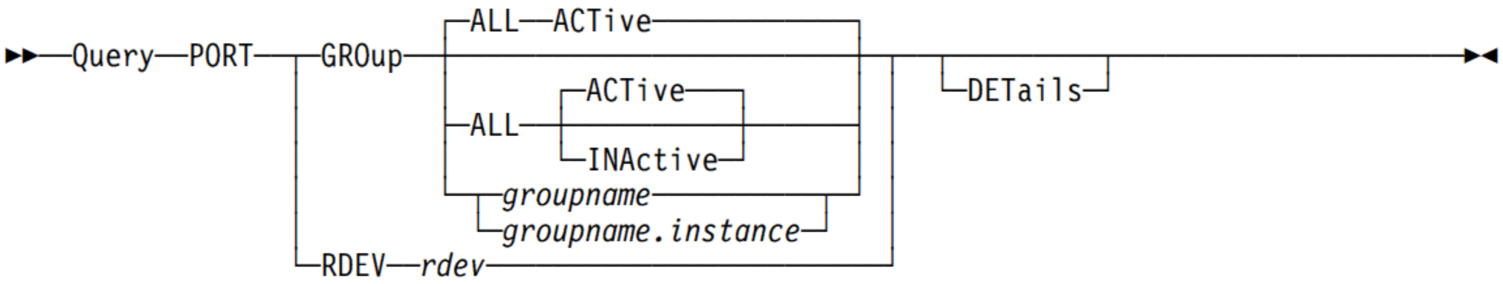
- CP QUERY VSWITCH ACTIVE
  - to find out which users are coupled
  - to find out which IP addresses are active

- CP QUERY NIC DETAILS
  - to find out if your adapter is coupled
  - to find out if your adapter is initialized
  - to find out if your IP addresses have been registered
  - to find out how many bytes/packets sent/received

Diagnostics: Discard Counters

Discard Counter	Uplink: QUERY VSWITCH ACTIVE	Guest NIC: QUERY NIC USER userid vdev
RX > 0 inbound	VSWITCH definition mismatch • Unused VLAN ID • VLAN UNAWARE on trunk	Packets are arriving faster than the guest can consume them
TX > 0 outbound	Overrun on the physical OSA. • Link is too slow compared to guests • Use faster OSA or link aggregation	• Unauthorized VLAN ID on virtual trunk port • Untagged frame on virtual trunk with NATIVE NONE • Guest configured as VLAN-aware with virtual access port • Overrun target guest
To reset	CP SET VSWITCH COUNTERS CLEAR	Resets when NIC is detached

Diagnostics: Port Group Verification



```
                          ┌─ALL──ACTive──────────────────────┐
►►──Query──PORT──┬──GROup─┼──────────────────────────────────┼─┬───────────┬──────────►◄
                 │        │     ┌─ACTive───┐                 │ │           │
                 │        ├─ALL─┼──────────┼─────────────────┤ │ └─DETails─┘
                 │        │     └─INActive─┘                 │ │
                 │        └─┬─groupname──────────────┬───────┘ │
                 │          └─groupname.instance─────┘         │
                 └──RDEV──rdev─────────────────────────────────┘
```

- ALL ACTIVE All port groups that are associated with a virtual switch
- ALL INACTIVE All port groups that are not associated with a virtual switch
- *groupname* or *groupname.instance*
  - The specified port group, optionally qualified by instance ID
- RDEV Information about the specified real device
- DETAILS Additional information
- See also SET VSWITCH … IVLPORT PING for a shared port group

References

— HELP command

`help sysconfig definvsw`	`DEFINE VSWITCH statement in SYSTEM CONFIG`
`help sysconfig vmlan`	`VMLAN statement in SYSTEM CONFIG`
`help define vswitch`	`CP DEFINE VSWITCH command`
`help cpset port`	`CP SET PORT GROUP command`
`help directory nicdef`	`NICDEF statement in user directory entry`

— Publications:

- z/VM CP Planning and Administration
- z/VM CP Command and Utility Reference
- z/VM Connectivity

Contact Information

Alan Altmark
Senior z/VM Engineer and Consultant

IBM Infrastructure
z/VM Development

IBM

1701 North Street
Endicott, NY 13760

Mobile 607 321 7556

Email: Alan_Altmark@us.ibm.com