#SHAREorg

SHARE

Networking with Linux on System z

Hans J. Picht – hans@de.ibm.com
IBM Corporation

Aug 7, 2012: 4:30 PM-5:45 PM
Grand Ballroom Salon C (Anaheim Marriott Hotel)

Session Number **11807**

SHARE in Anaheim 2012

Hans-Joachim Picht, Linux on System z Initiatives & Linux Champion EMEA <hans@de.ibm.com>

Networking with Linux on System z

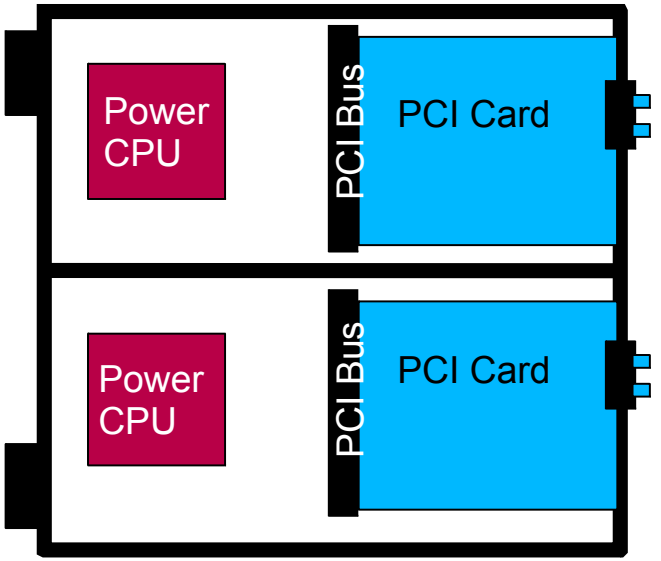
Session 11807
Tuesday, August 7, 2012: 4:30 PM-5:45 PM
Grand Ballroom Salon C (Anaheim Marriott Hotel)

Agenda

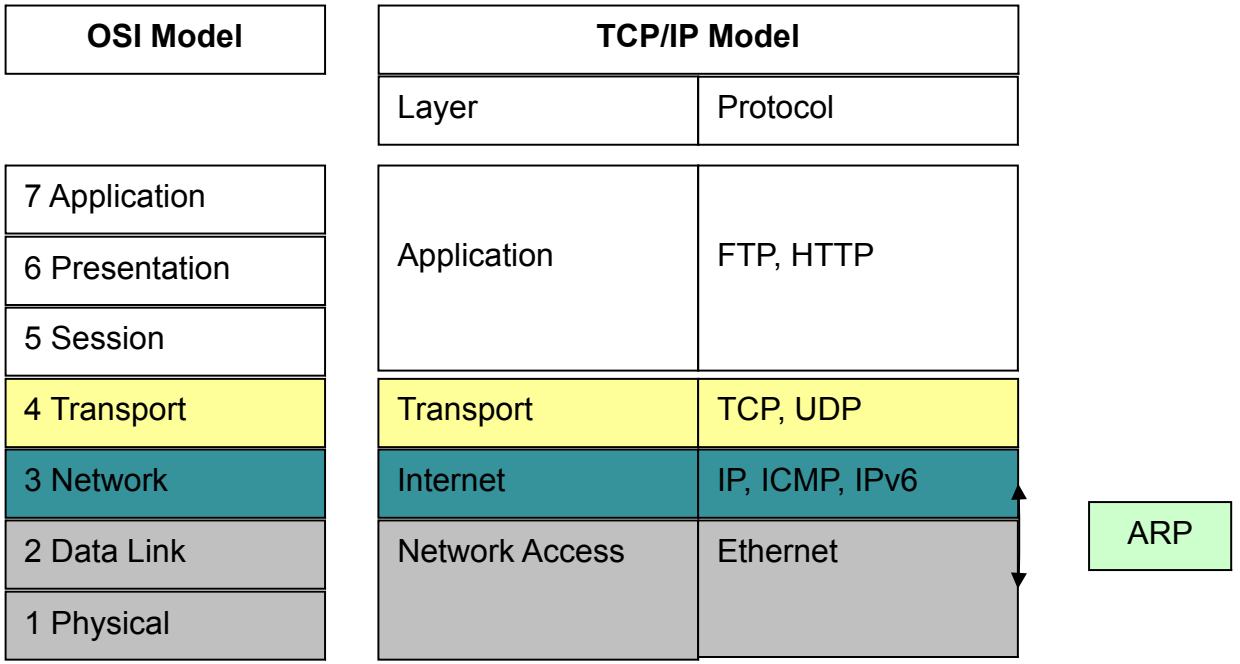
Network Connections

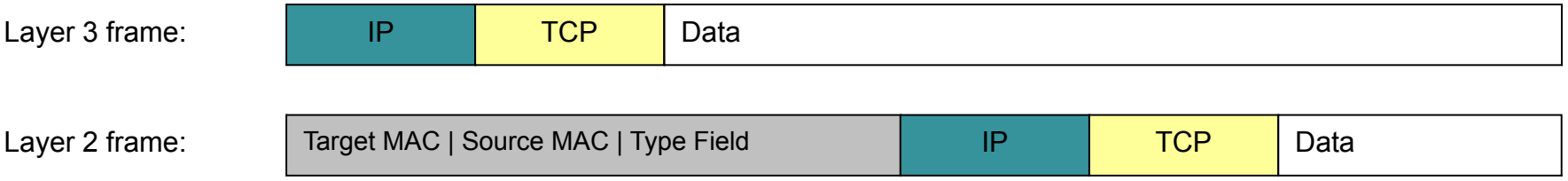
Primary Network Device: OSA Express

- Open Systems Adapter-Express (OSA-Express) provides connectivity support to the following LAN types: 1000BASE-T Ethernet (10/100/1000 Mbps), 1 Gbps Ethernet, 10 Gbps Ethernet
- 'Integrated Power computer' with network daughter card
- Shared between up to 640 OSA devices
- Three devices numbers (ccw devices) per OSA device:
  - Read device (control data ← OSA)
  - Write device (control data → OSA)
  - Data device (network traffic)
- OSA Address Table: which OS image has which IP address
- Network traffic Linux ↔ OSA, either
  - IP (layer3 mode)
    - One MAC address for all stacks
    - OSA handles ARP (Address Resolution Protocol)
  - Ethernet / data link layer level (layer2 mode)



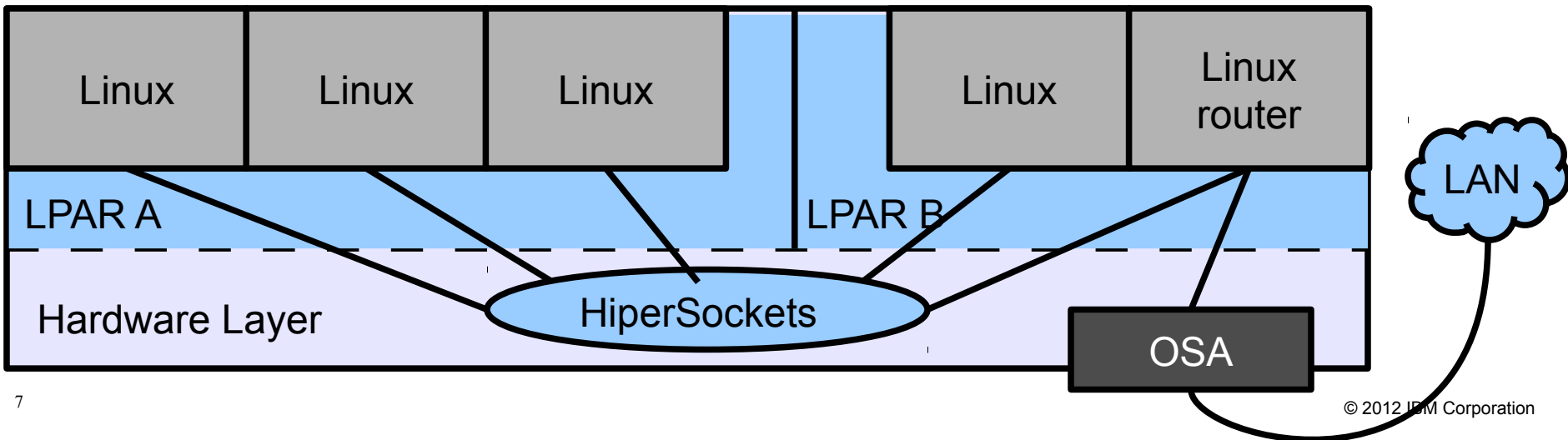
QETH Layer 3 vs Layer 2 mode





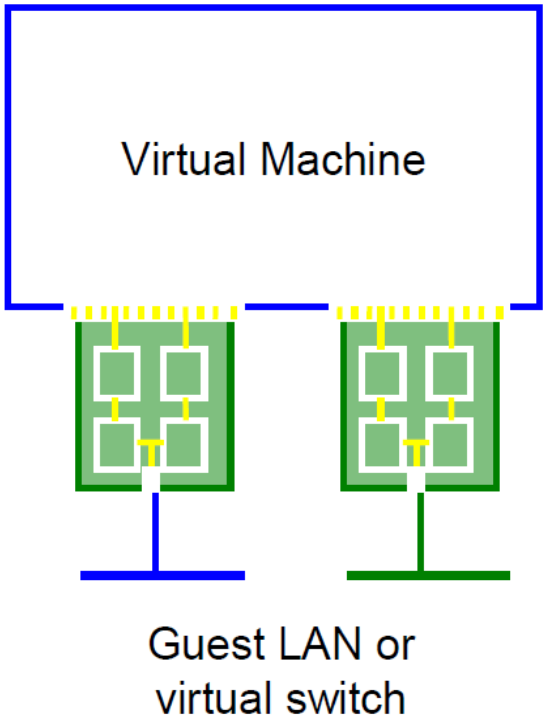
System z Hipersockets

- Connectivity within a central processor complex without physical cabling
- Internal Queued Input/Output (IQDIO) at memory speed
- Licensed Internal Code (LIC) function emulating DataLink Layer of an OSA-device (internal LAN)
- 4 different MTU sizes supported:
  - 8KB, 16KB, 32KB, 56KB
- Support of
  - Broadcast, VLAN, Ipv6, Layer2 (with z10)



Virtual Network Interface Card (NIC)

- A simulated network adapter
  - OSA-Express QDIO
  - HiperSockets
  - Must match LAN type
- Usually 3 devices per NIC
- Provides access to Guest LAN or VSWITCH
- Created by directory or *CP DEFINE NIC*



z/VM Guests (Linux, zVSE, ...)

```
DEF NIC 600 TYPE QDIO
COUPLE 600 SYSTEM VSWITCH1
```

Virtual Switch

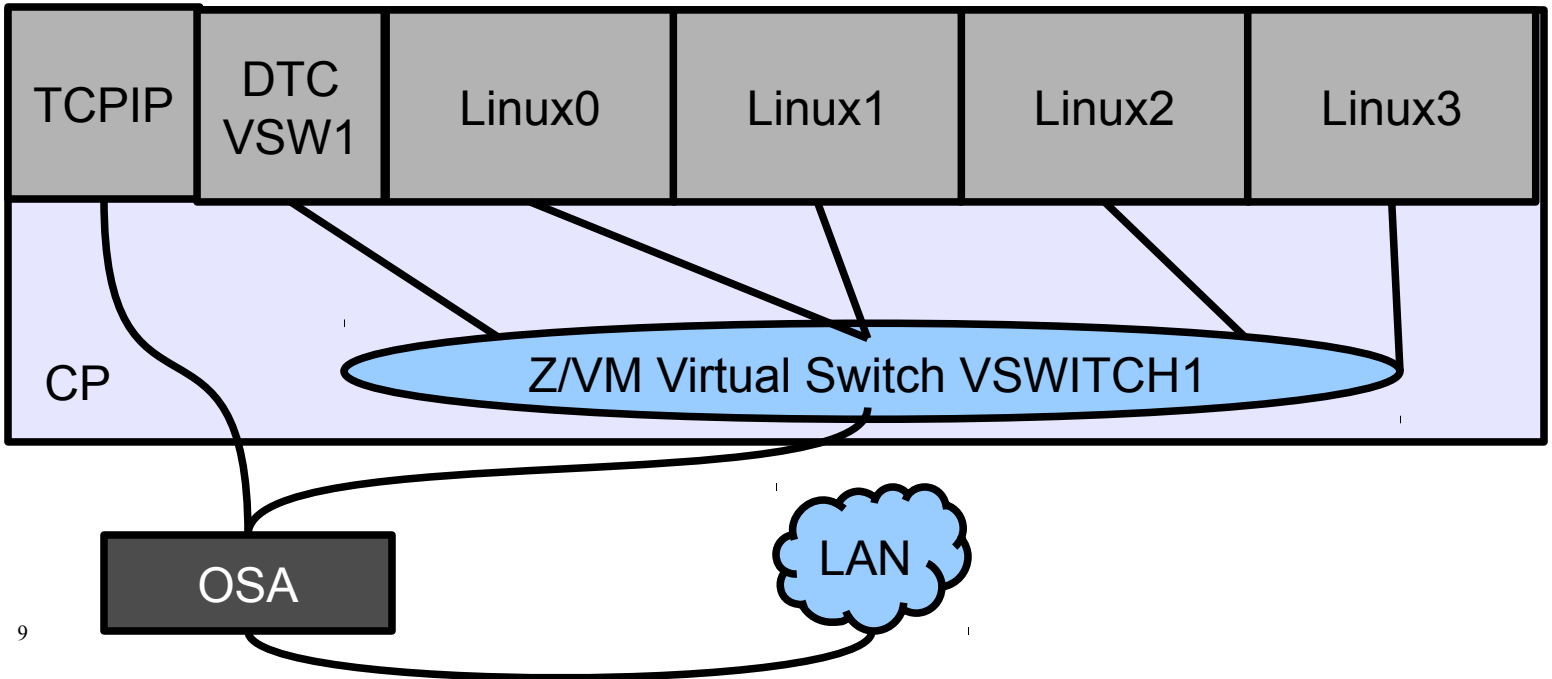
- Create simulated Layer 2 or 3 switch device
- VM access control and VLAN authorization
- Create ports
- Connect NIC to Virtual Switch (LAN Segment)
- Full MAC address management
  (generation and assignment)
- 1-n VSWITCHs per z/VM Image

Create VSWITCH from PRIVCLASS B User ID

```
DEF VSWITCH VSWITCH1 ETHERNET
SET VSWITCH VSWITCH1 GRANT {user ID}
```

From Linux Virtual Machines

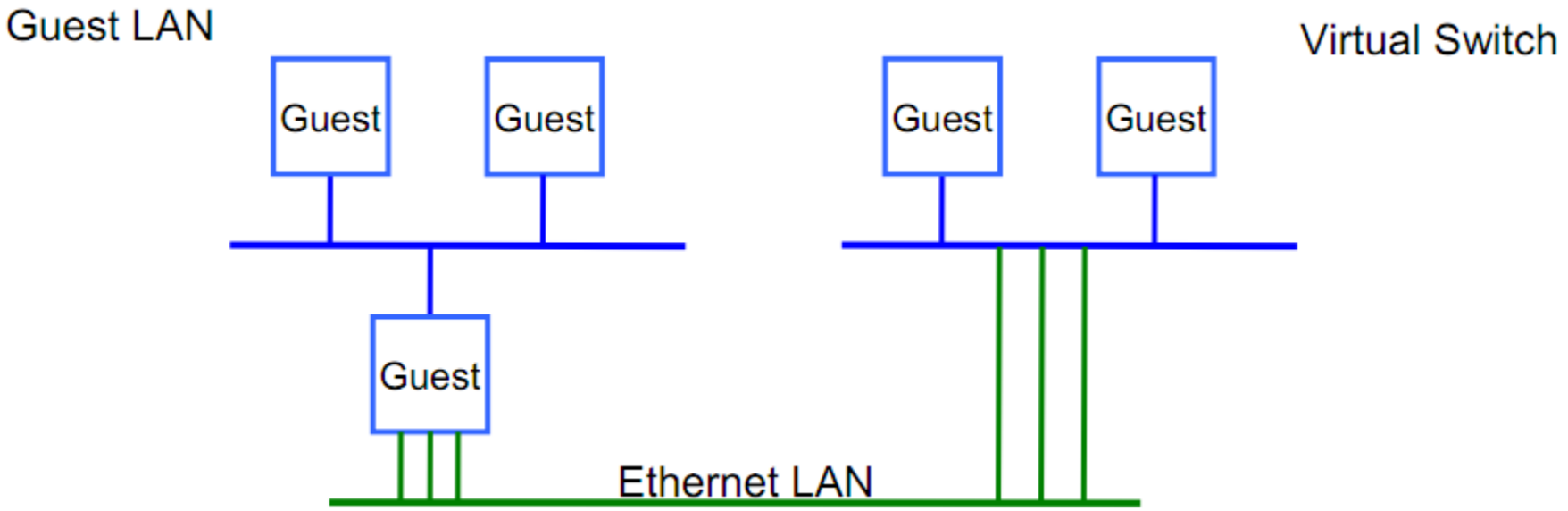
```
DEF NIC 600 TYPE QDIO
COUPLE 600 SYSTEM VSWITCH1
```



GuestLAN

- A simulated LAN
- Ethernet: IPv4 and IPv6
- HiperSockets: IPv4
- Unicast, Multicast, and Broadcast
- No built-in connection to outside network
- As many as you want
- Owned by system or individual user
- Is not a device - it is a system object
- Created in SYSTEM CONFIG, directory, or by CP
- DEFINE LAN command

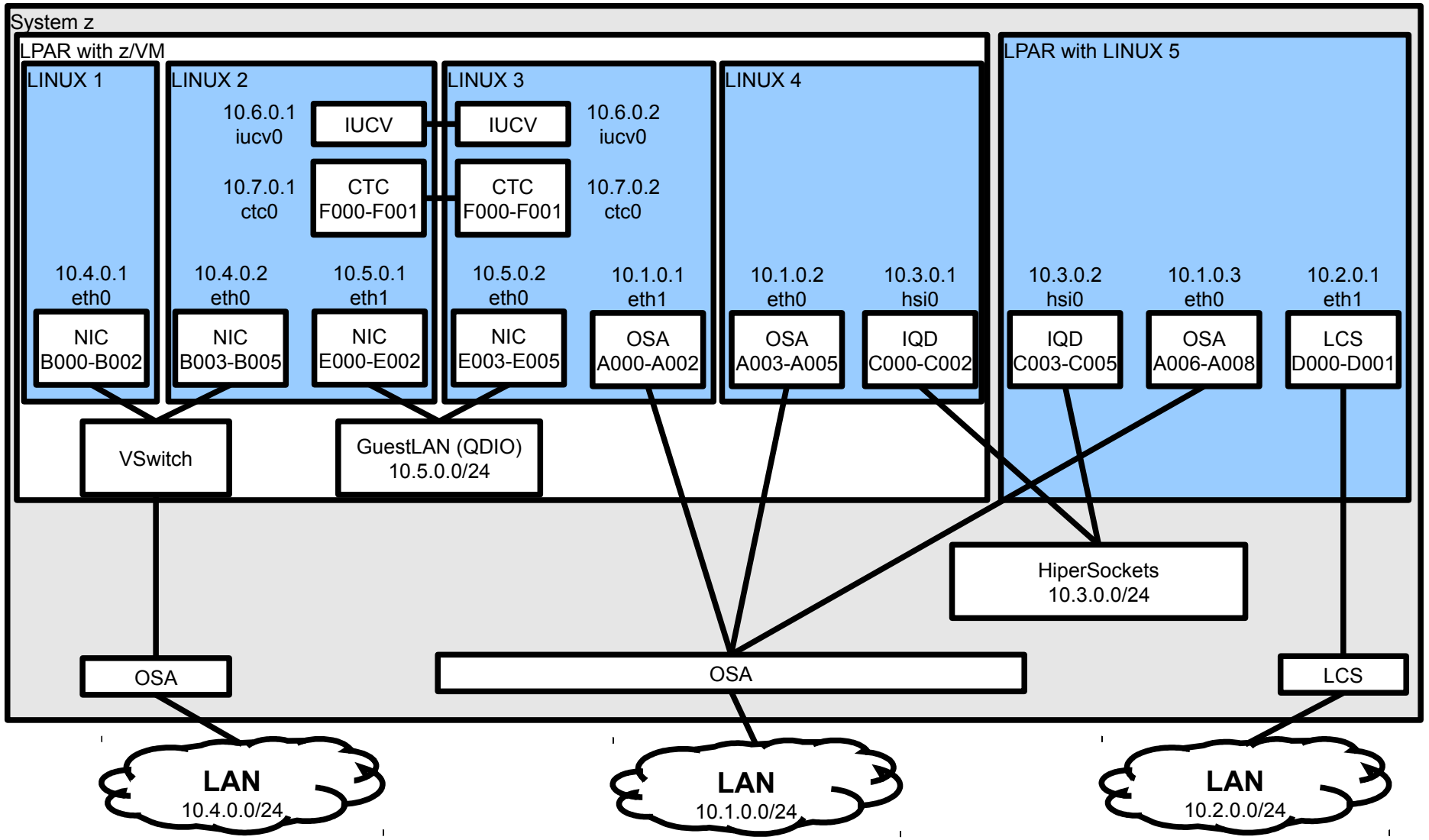
Guest LAN vs. Virtual Switch



- Virtual router is required
- Different subnets
- External router awareness
- Guest-managed failover

- No virtual router
- Same subnets
- Transparent bridge
- CP-managed failover

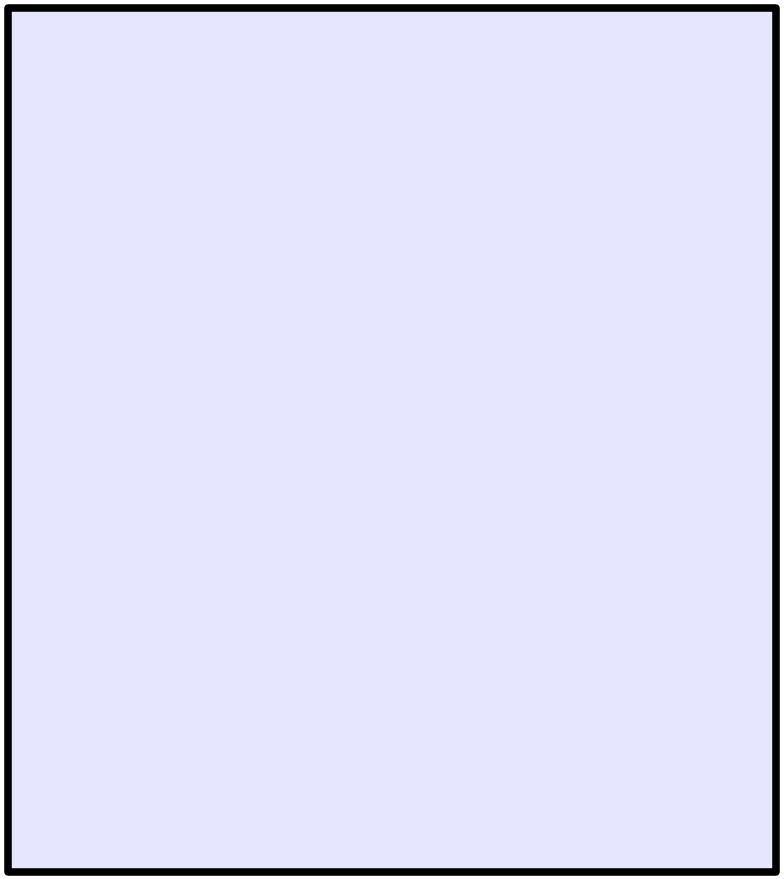
Network Example



Where is the Problem?

Where is the Problem?

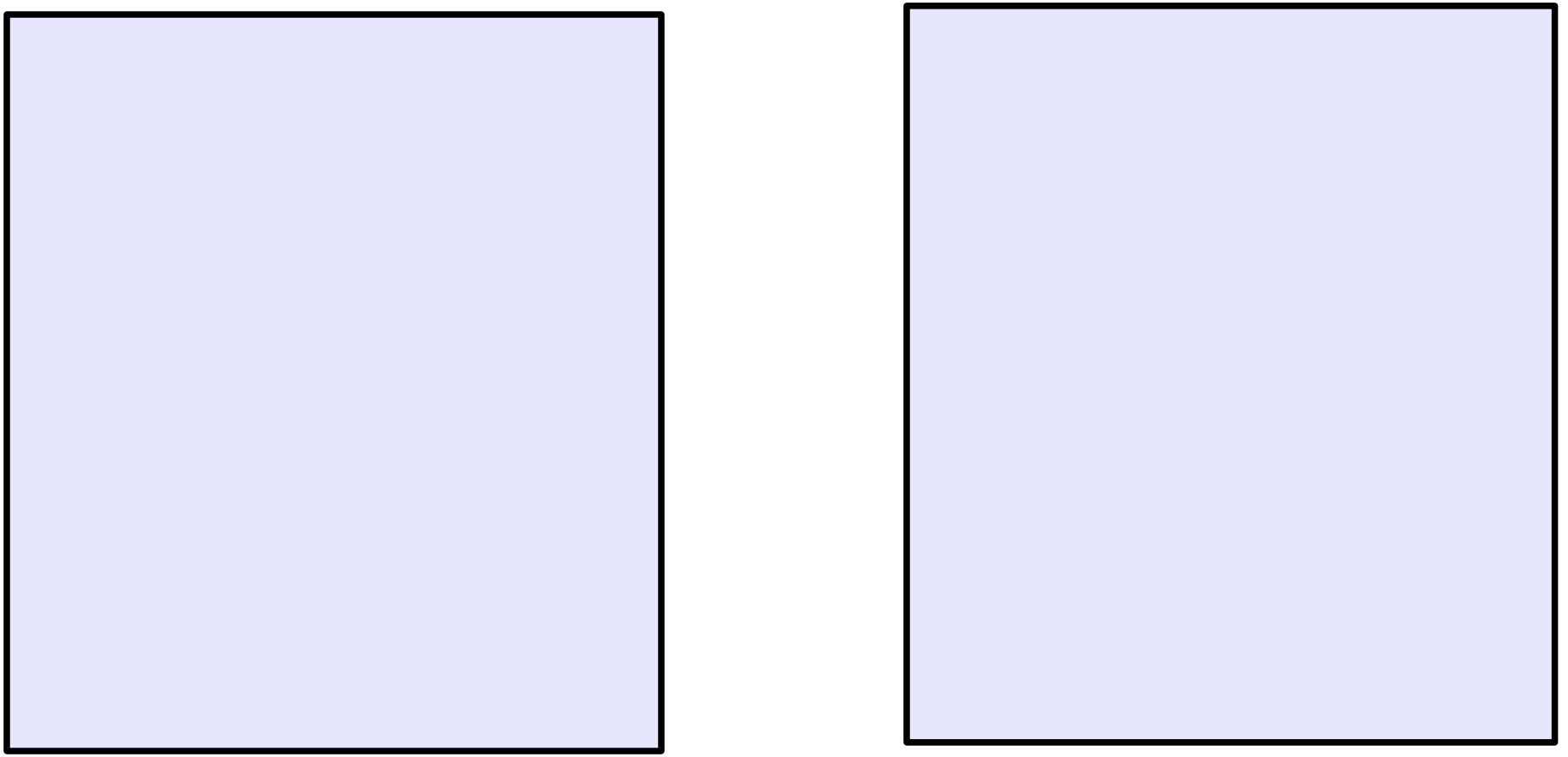
Site A



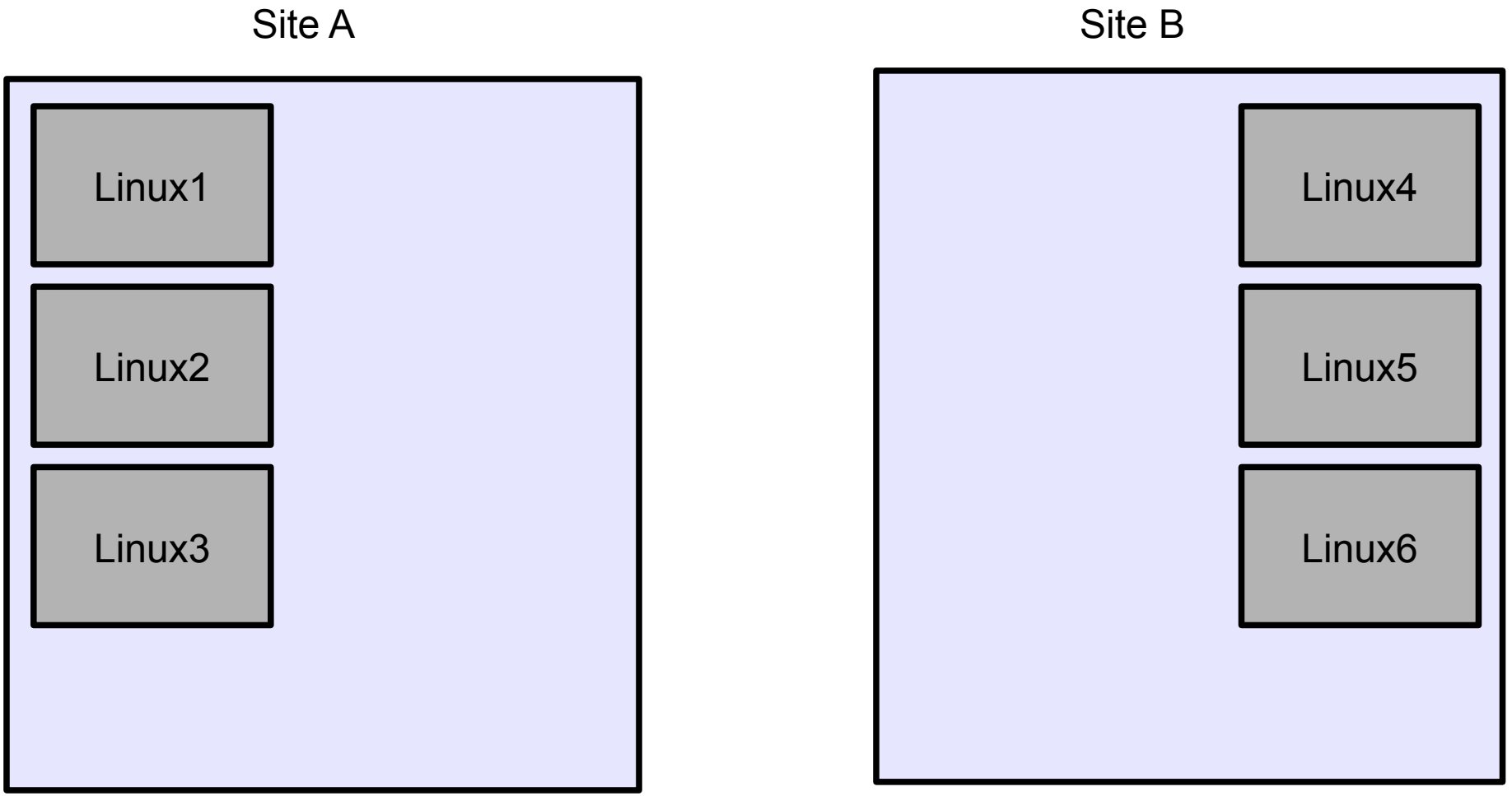
Where is the Problem?

Site A

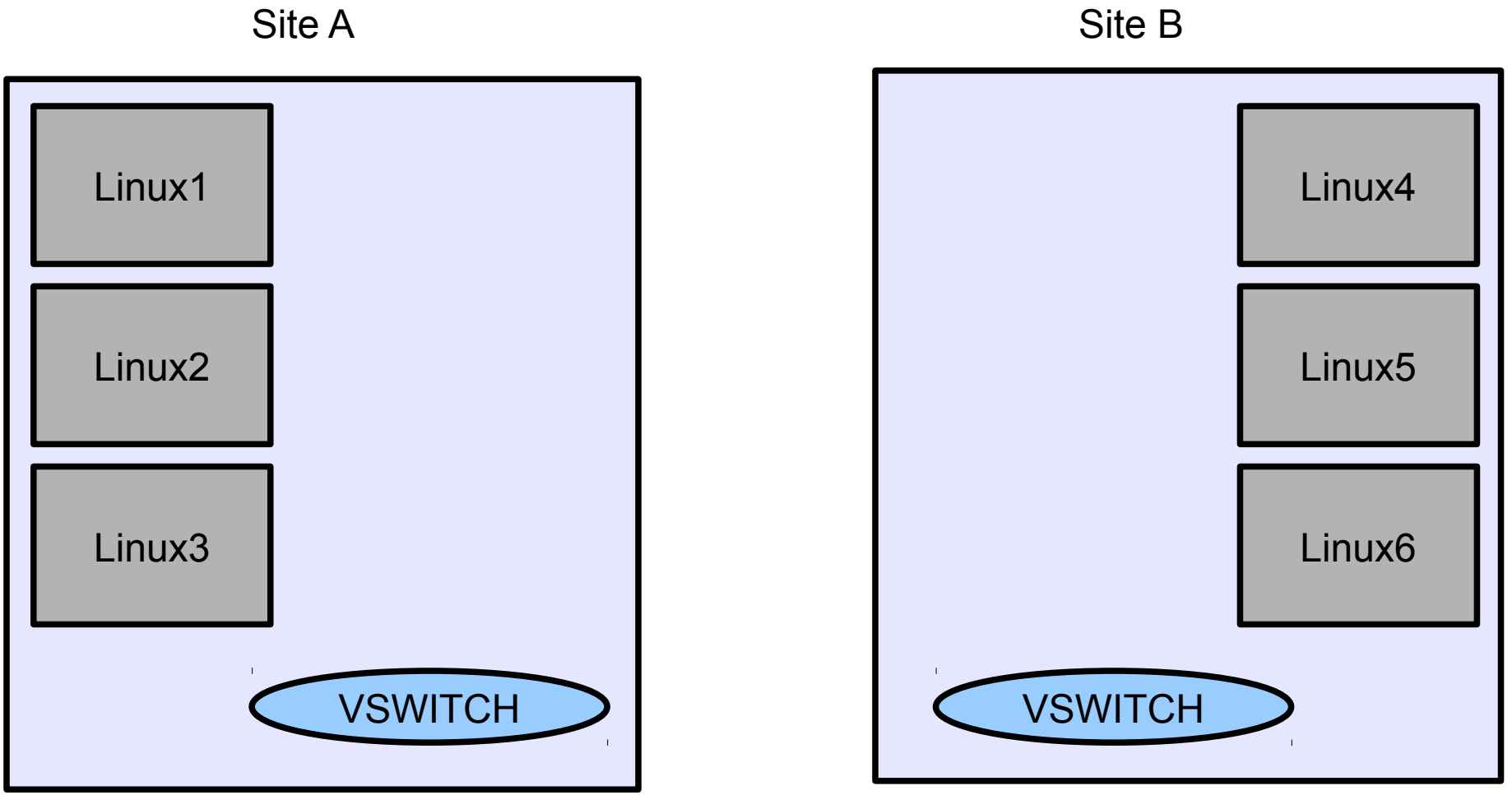
Site B



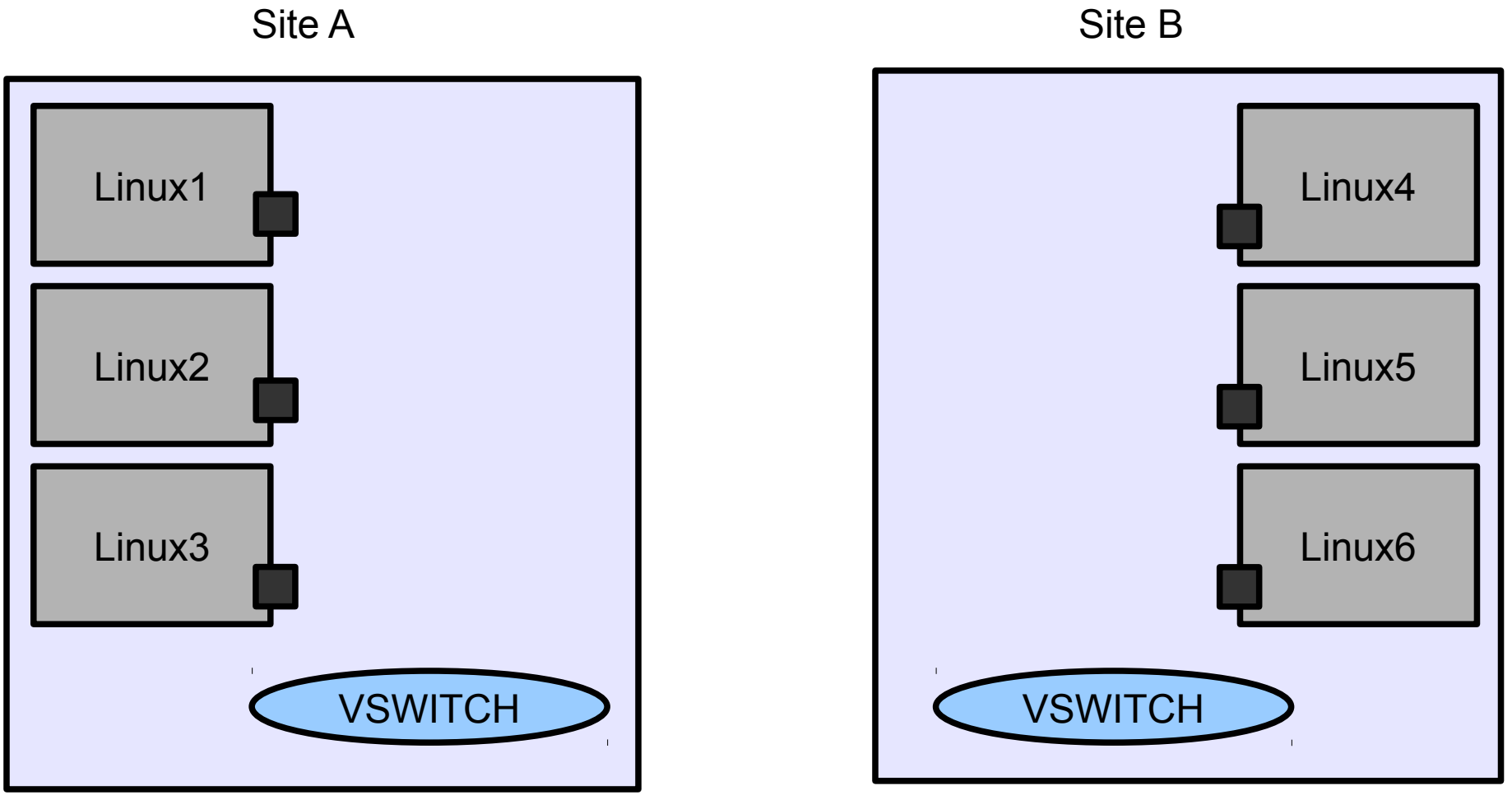
Where is the Problem?



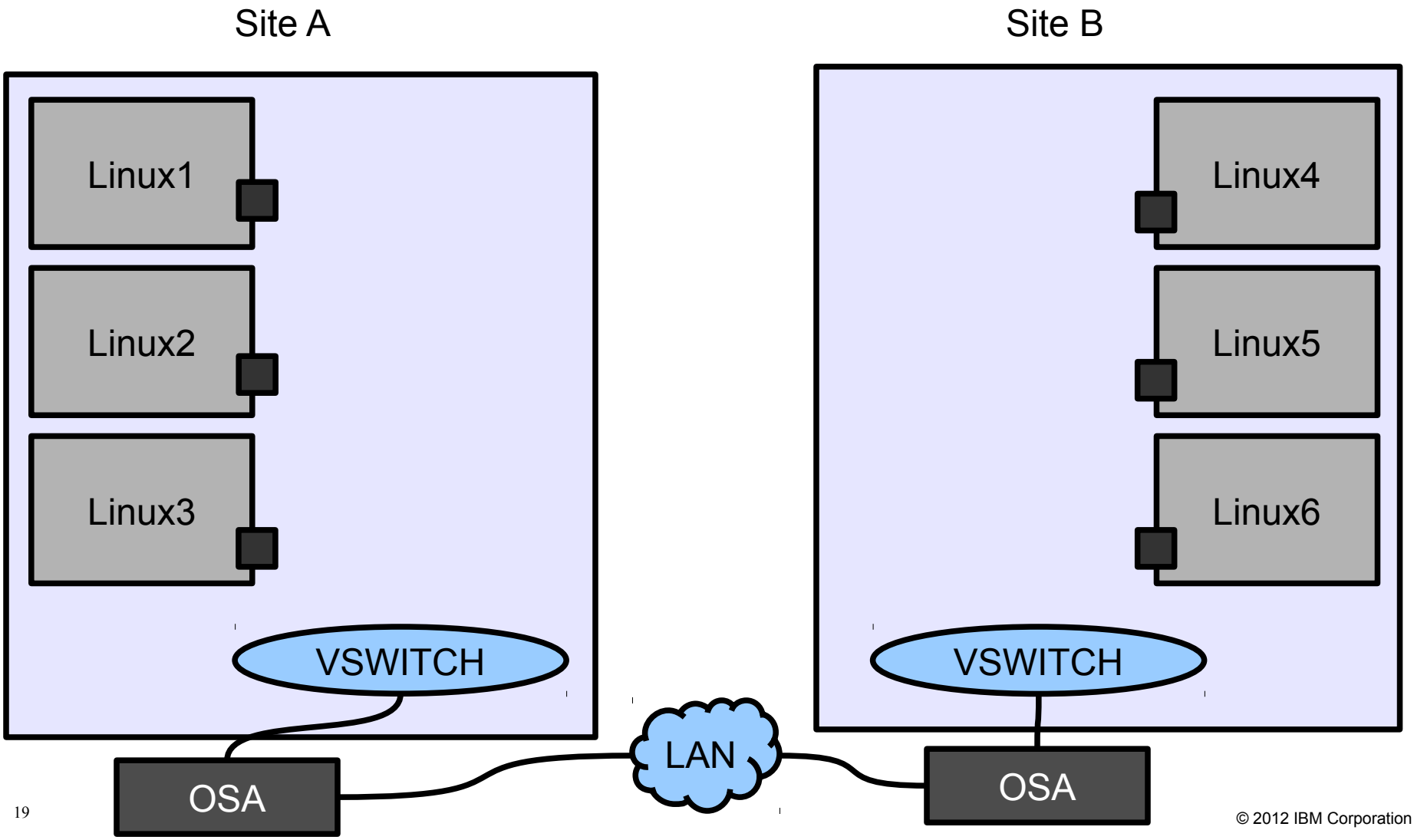
Where is the Problem?



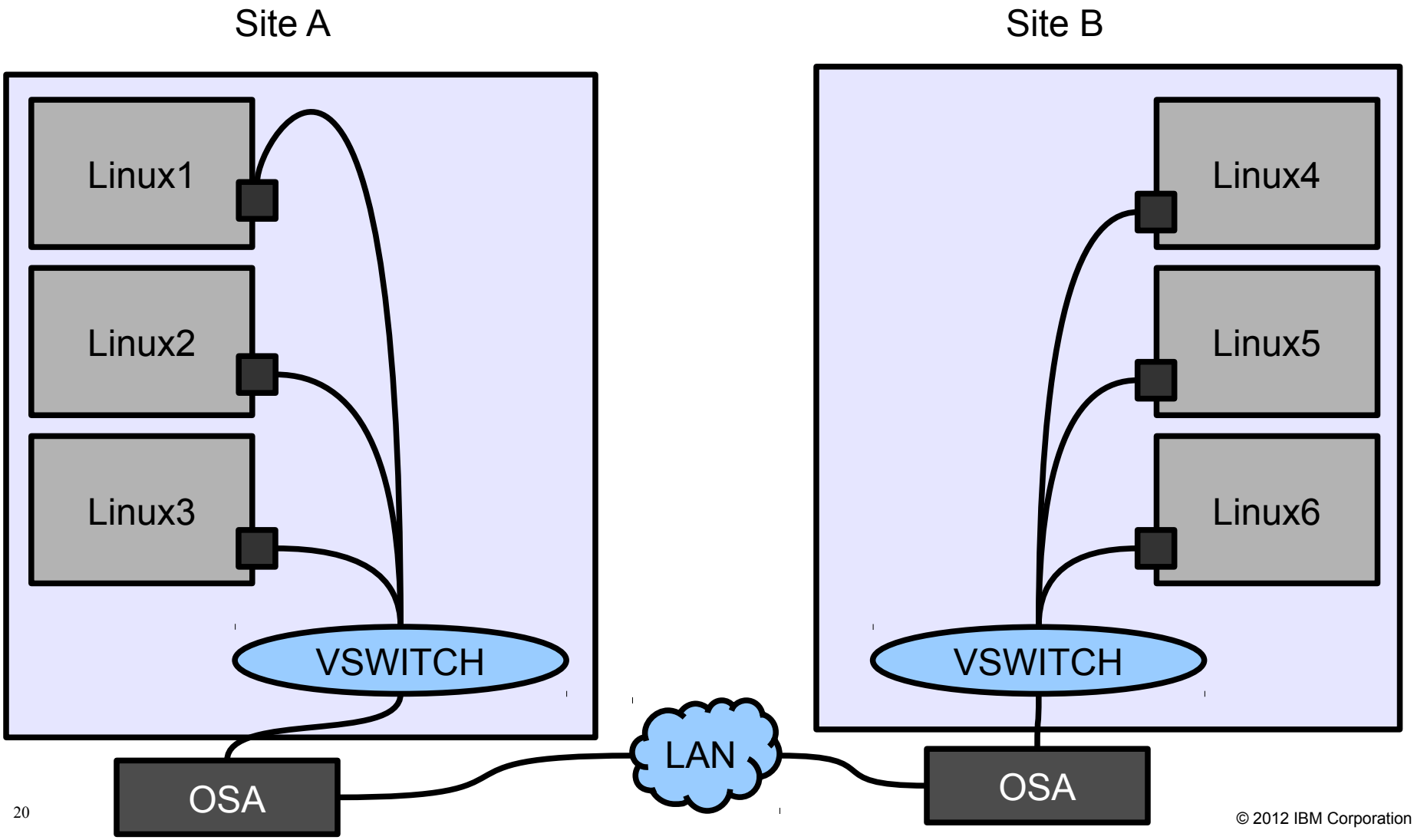
Where is the Problem?

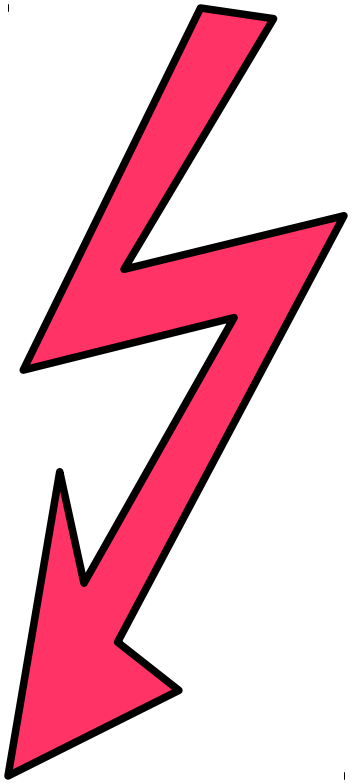


Where is the Problem?



Where is the Problem?

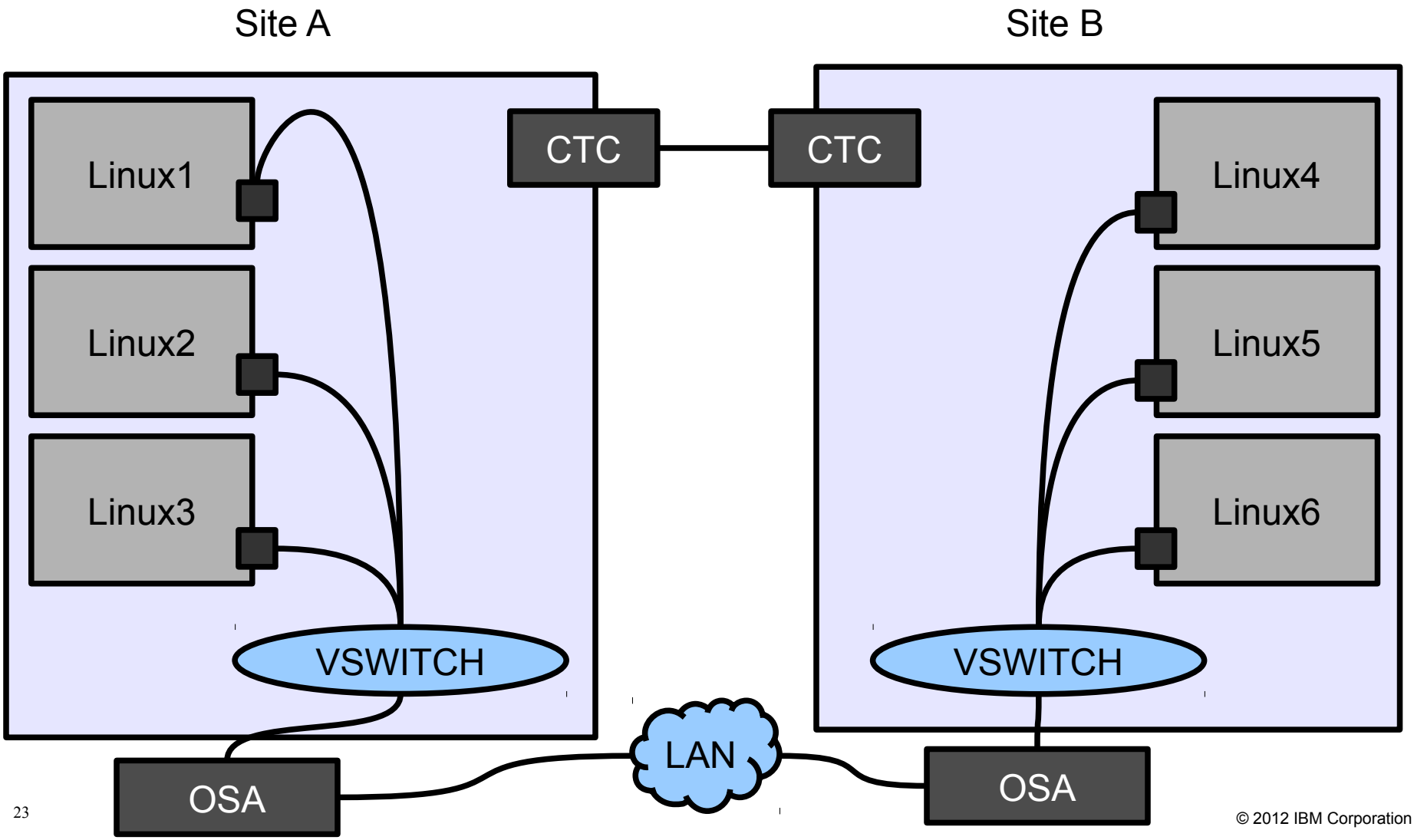




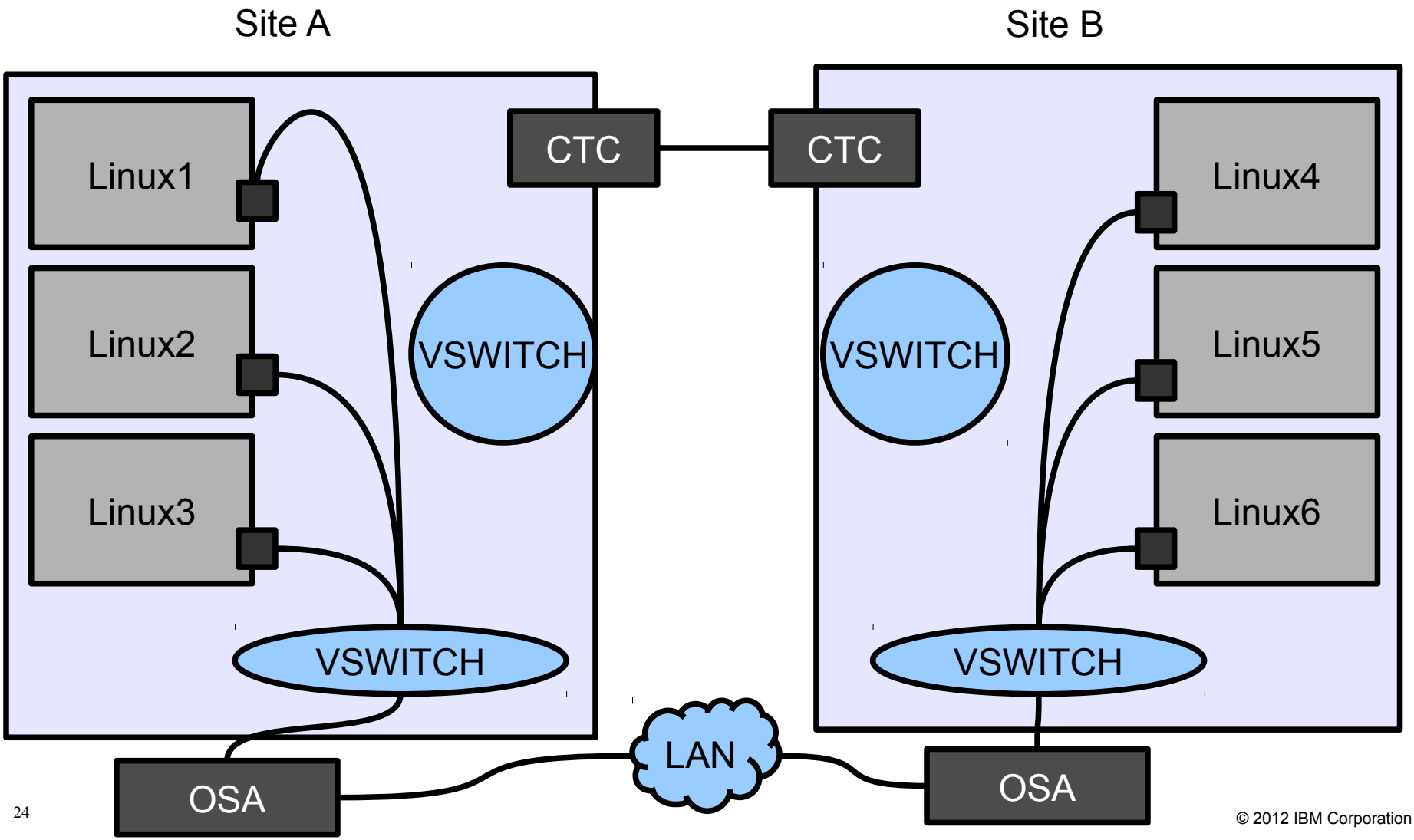
Everything is working fine...but the OSA cards are shared with the z/OS and the Linux TSM backup takes forever....

...So the Admin decides to implement a CTC connection for the backup...

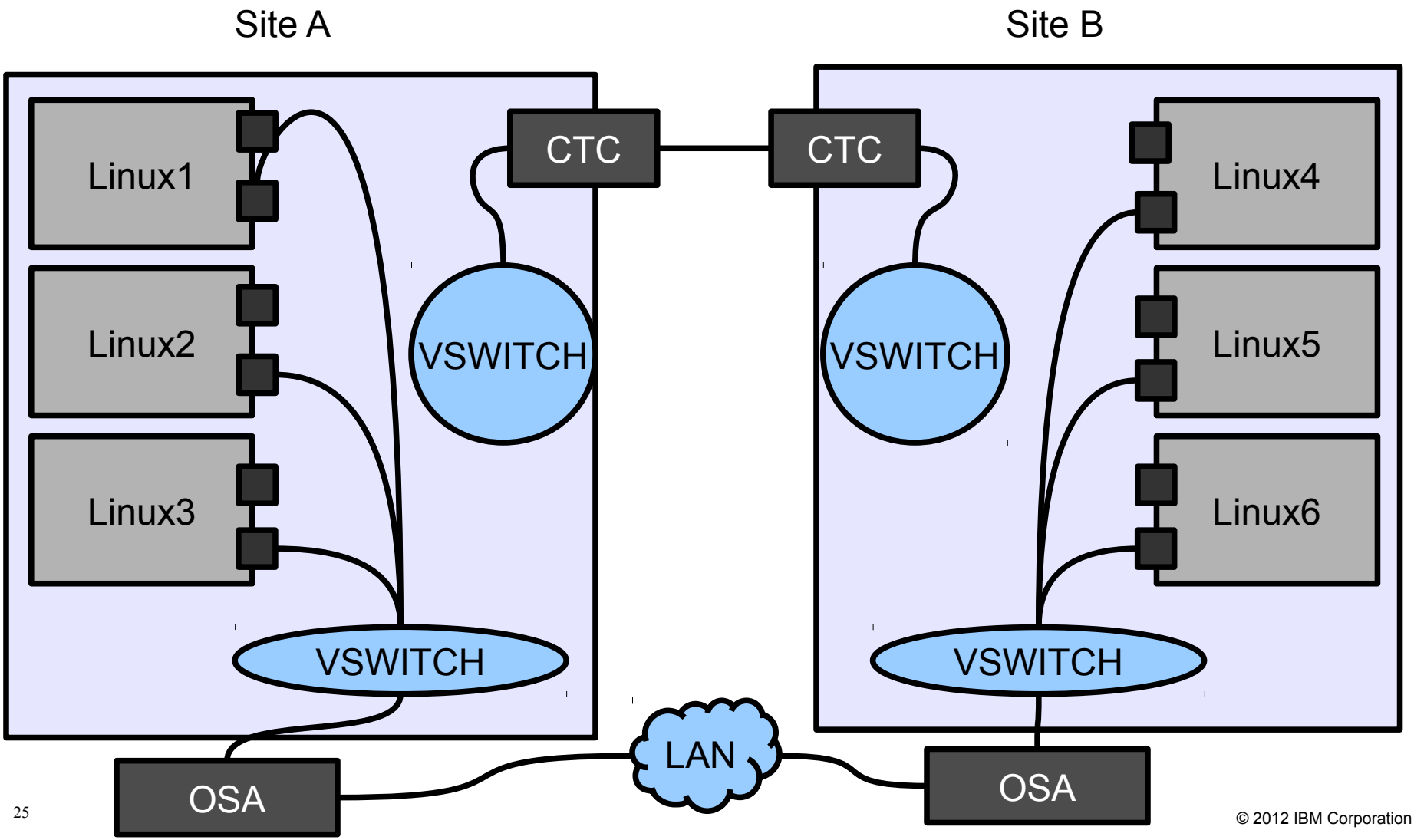
Where is the Problem?



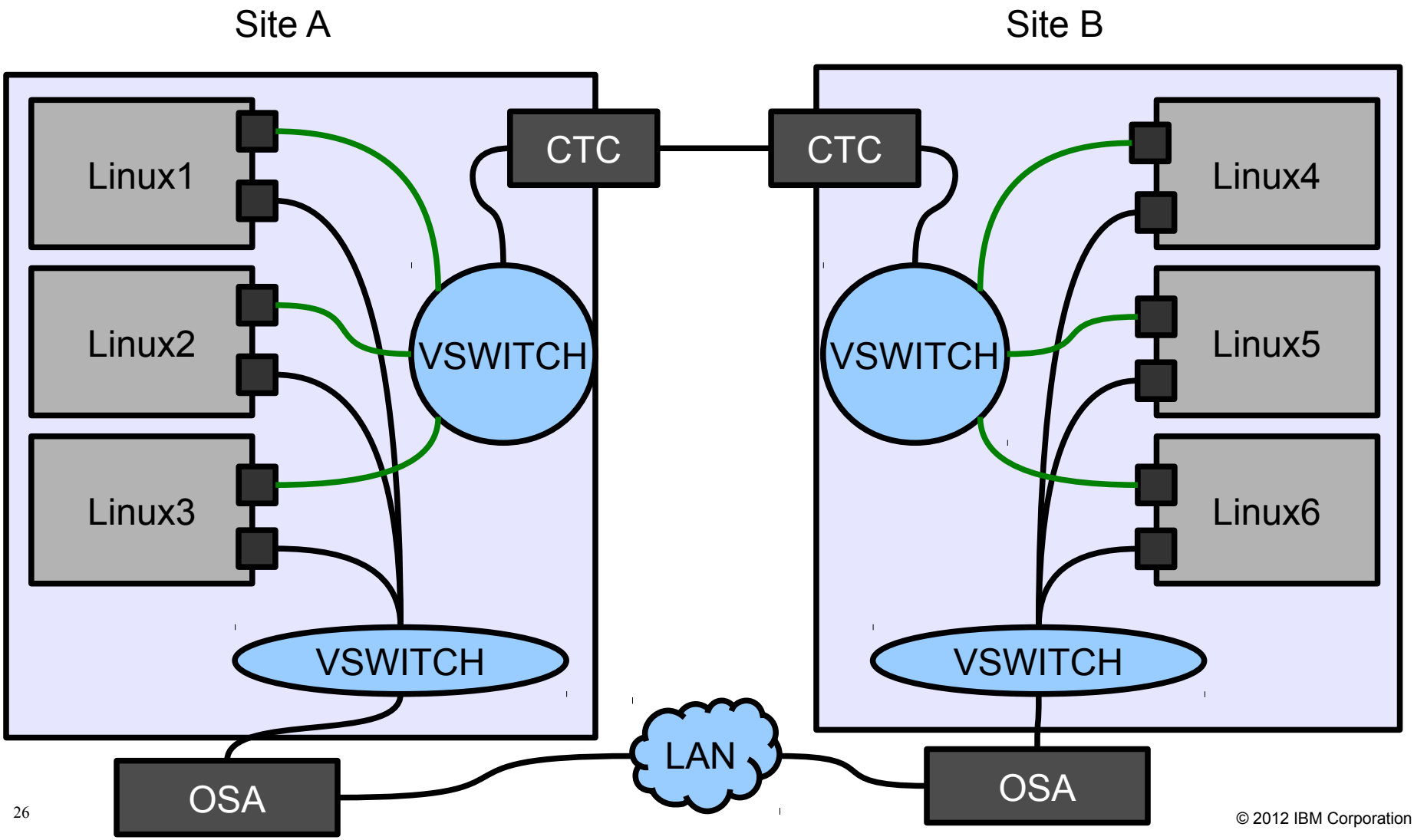
Where is the Problem?



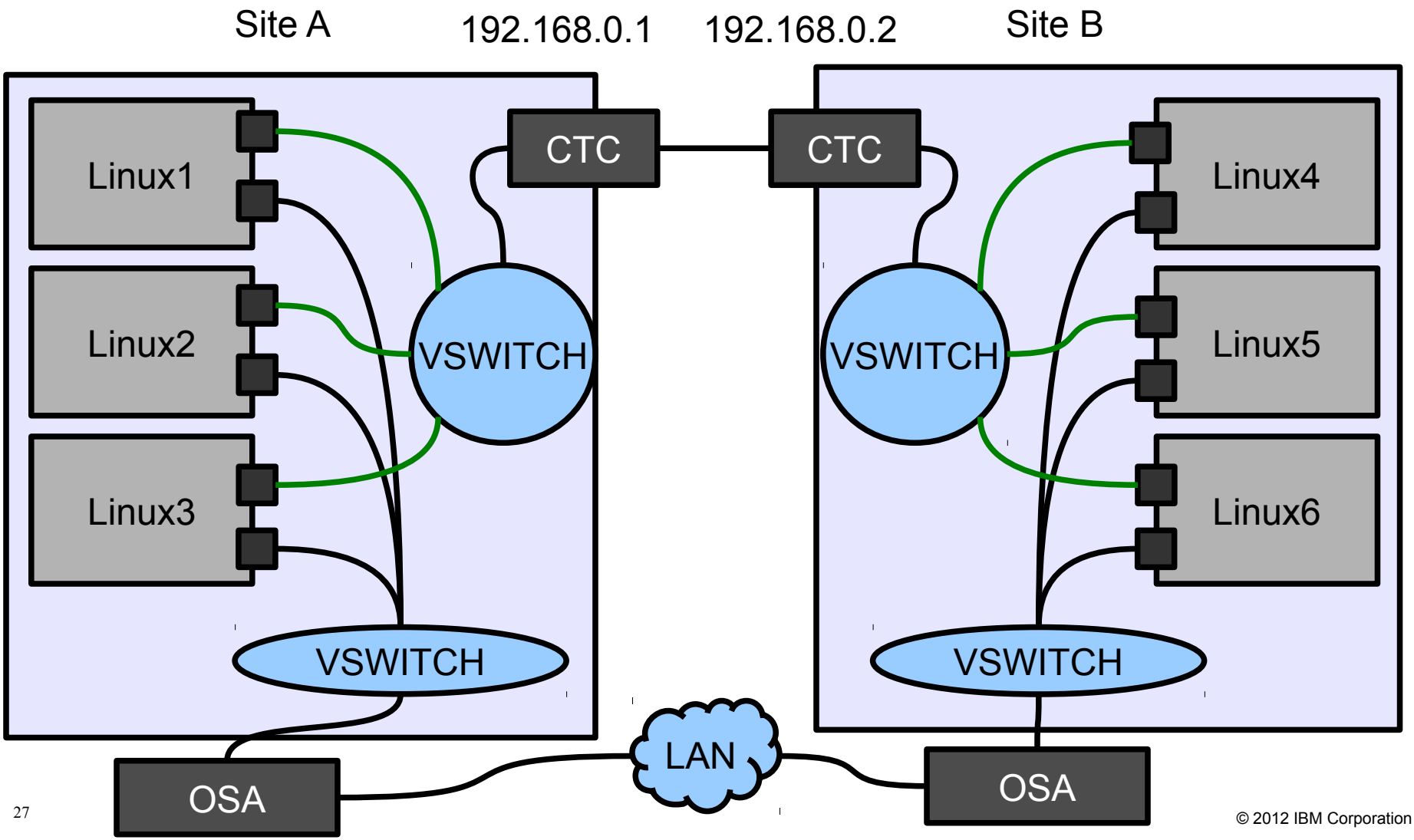
Where is the Problem?



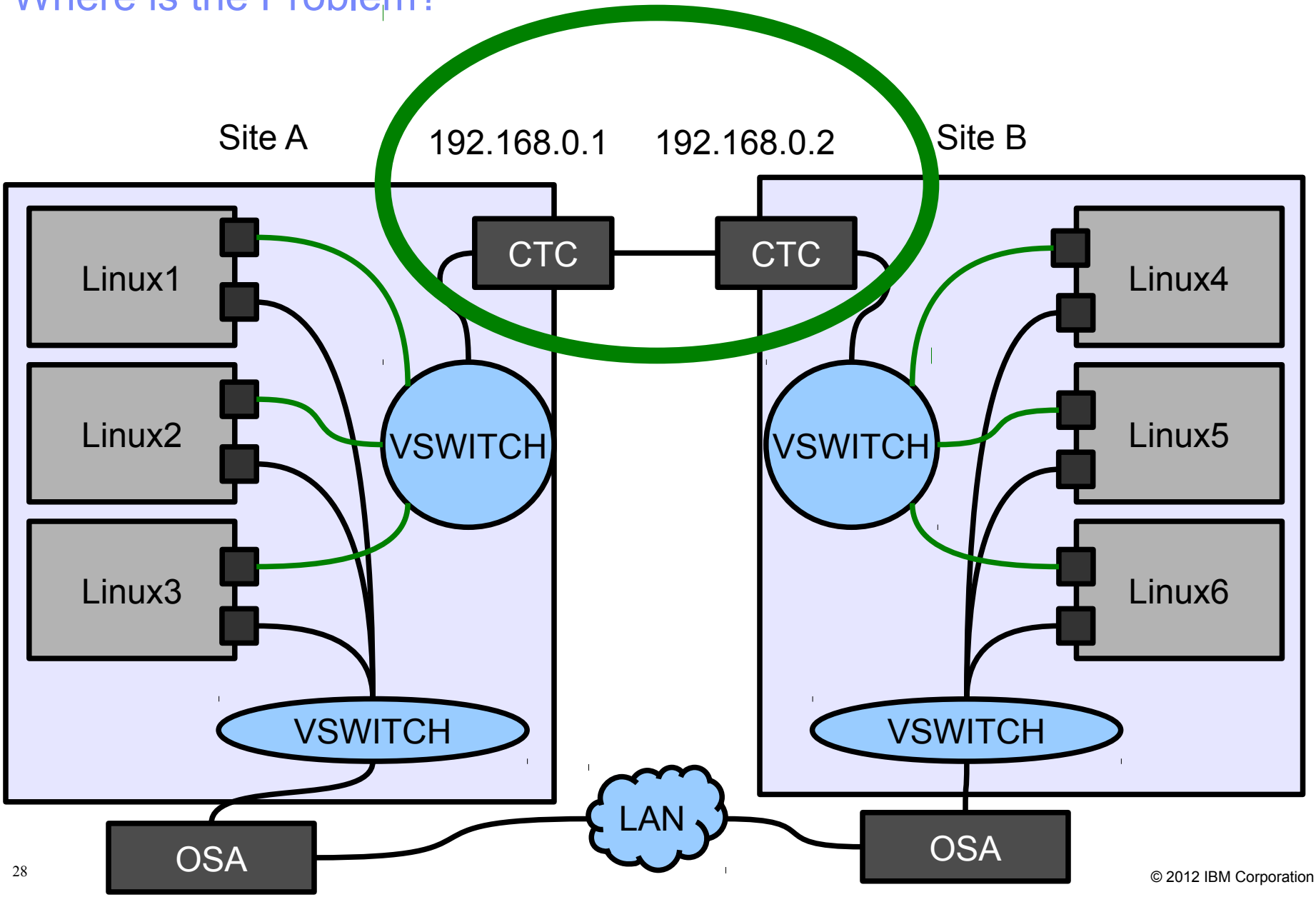
Where is the Problem?



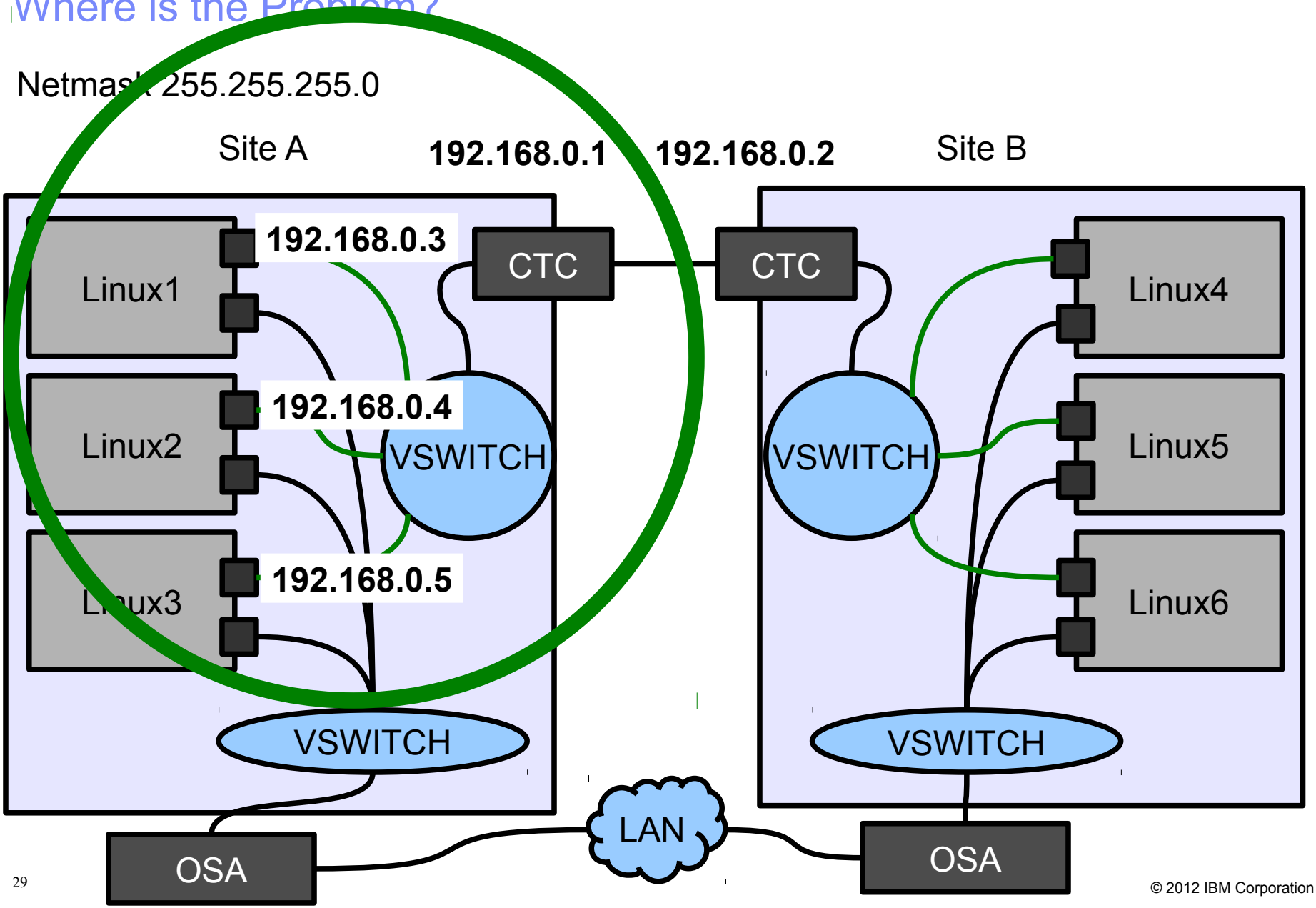
Where is the Problem?



Where is the Problem?

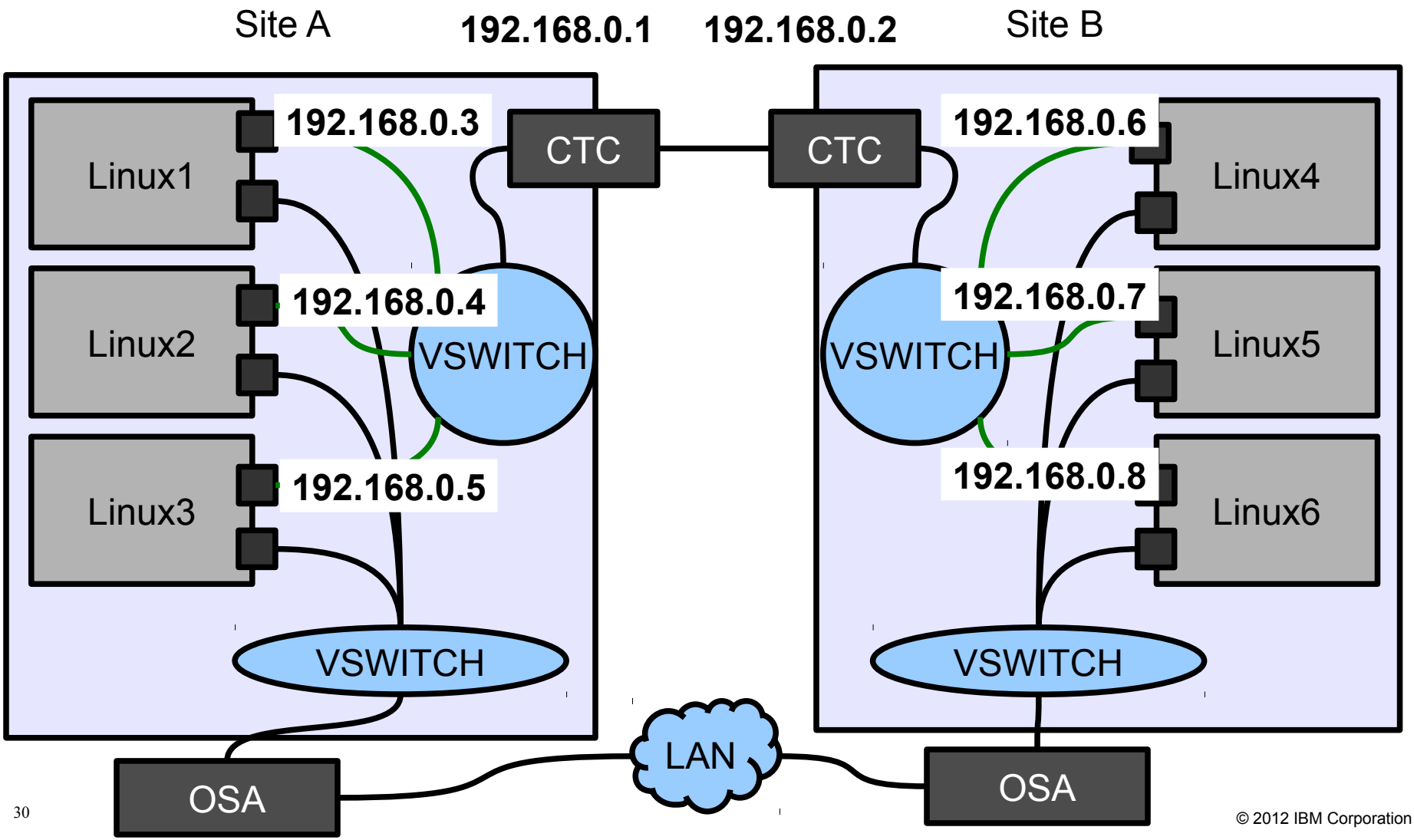


Where is the Problem?

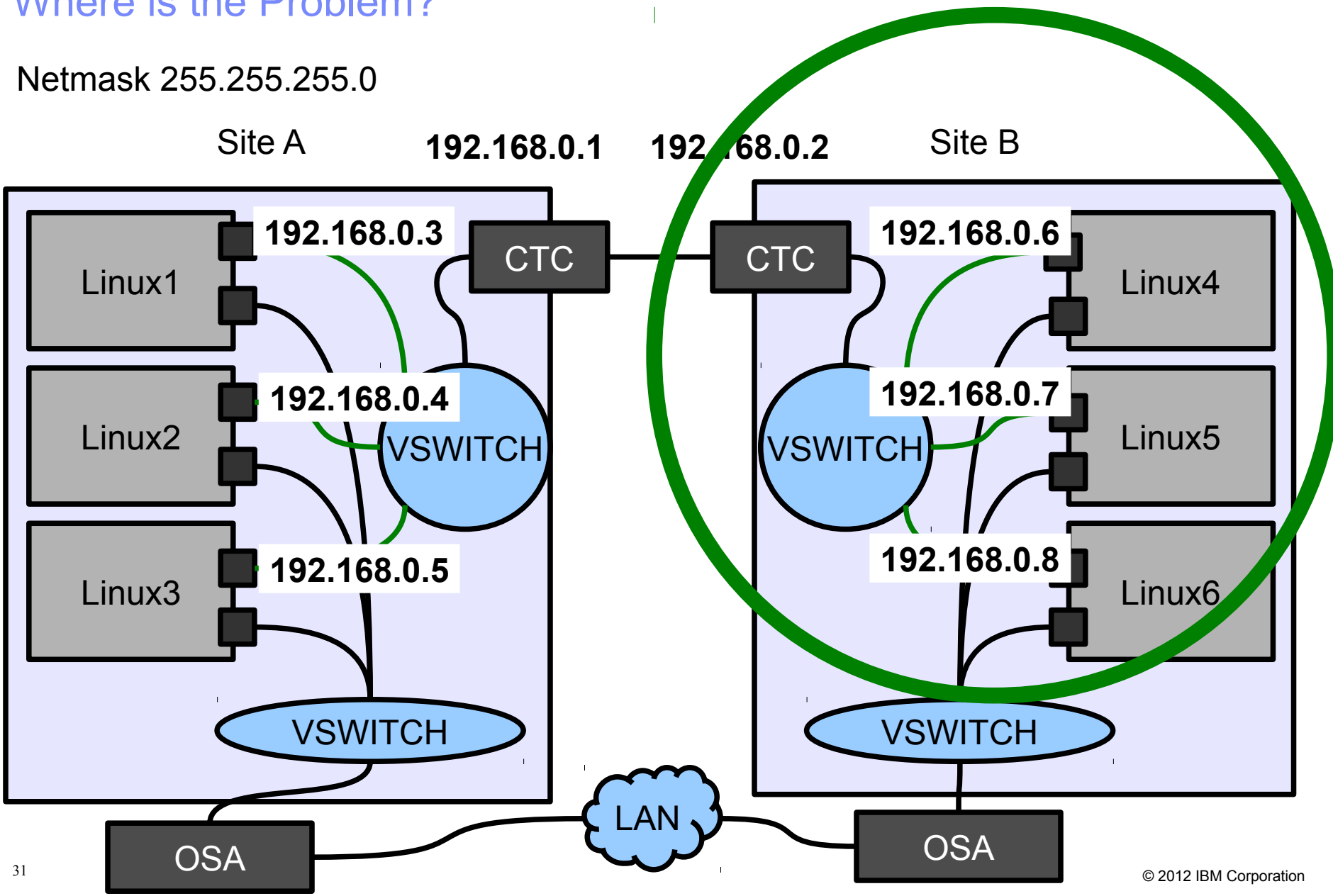


Where is the Problem?

Netmask 255.255.255.0

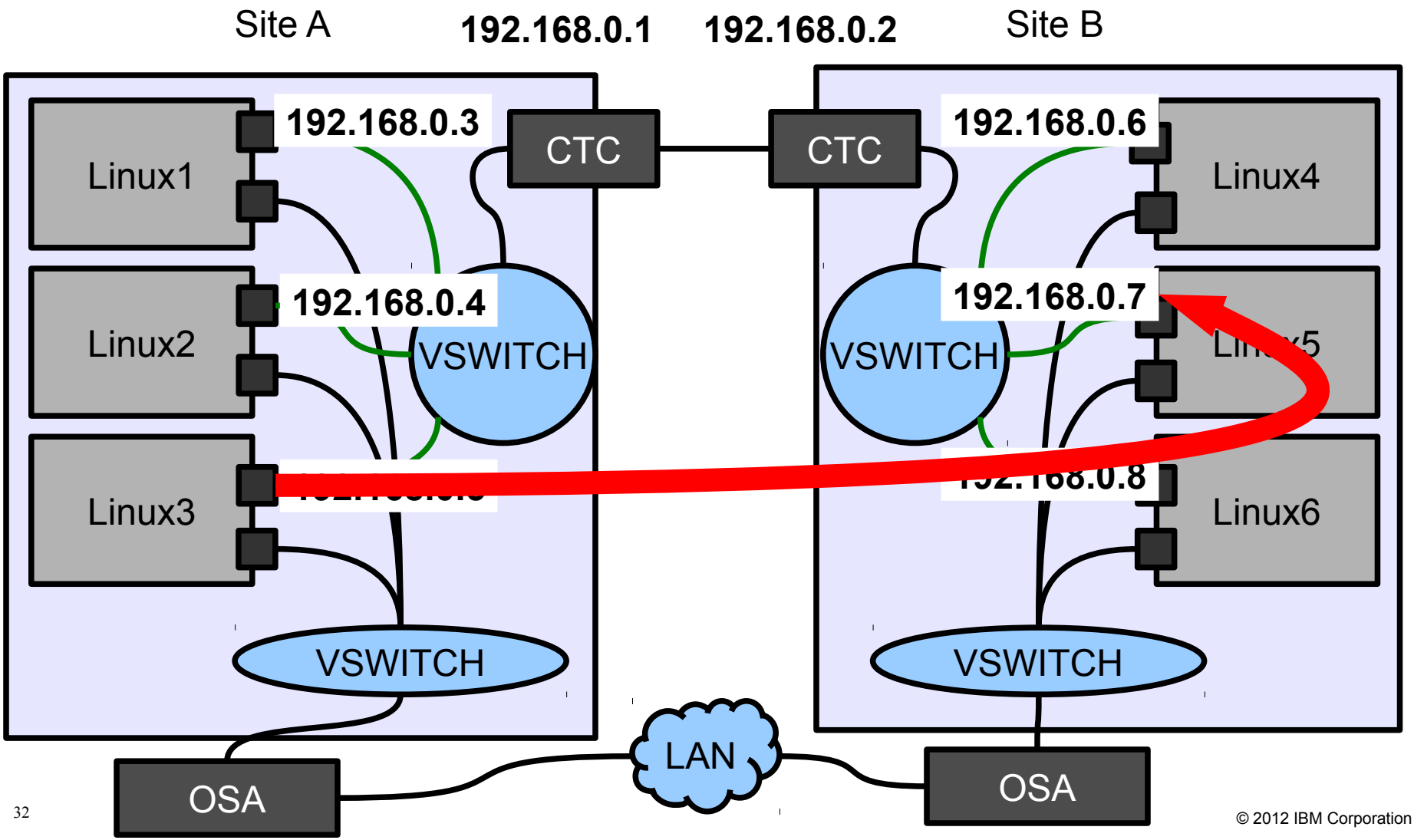


Where is the Problem?



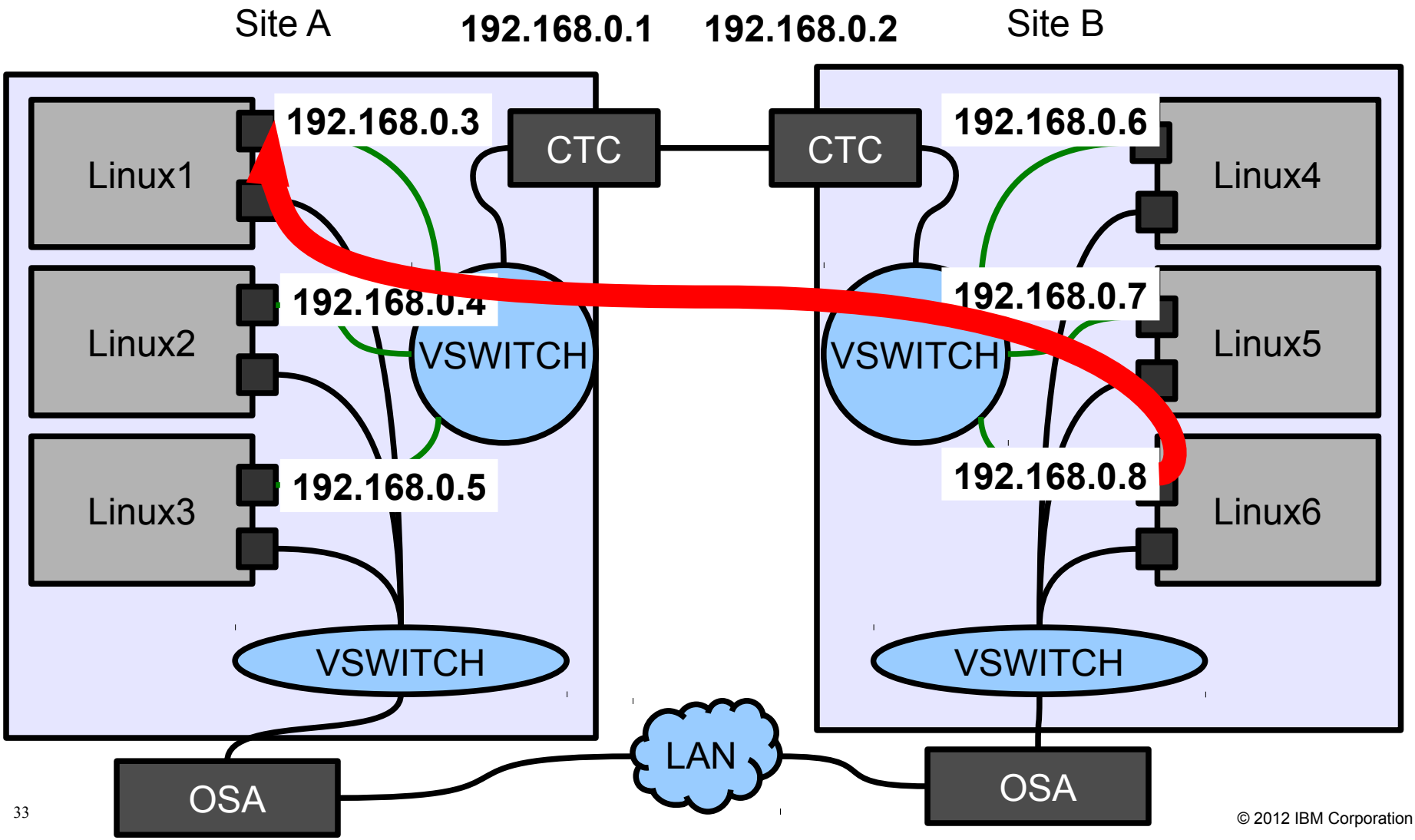
Where is the Problem?

Netmask 255.255.255.0

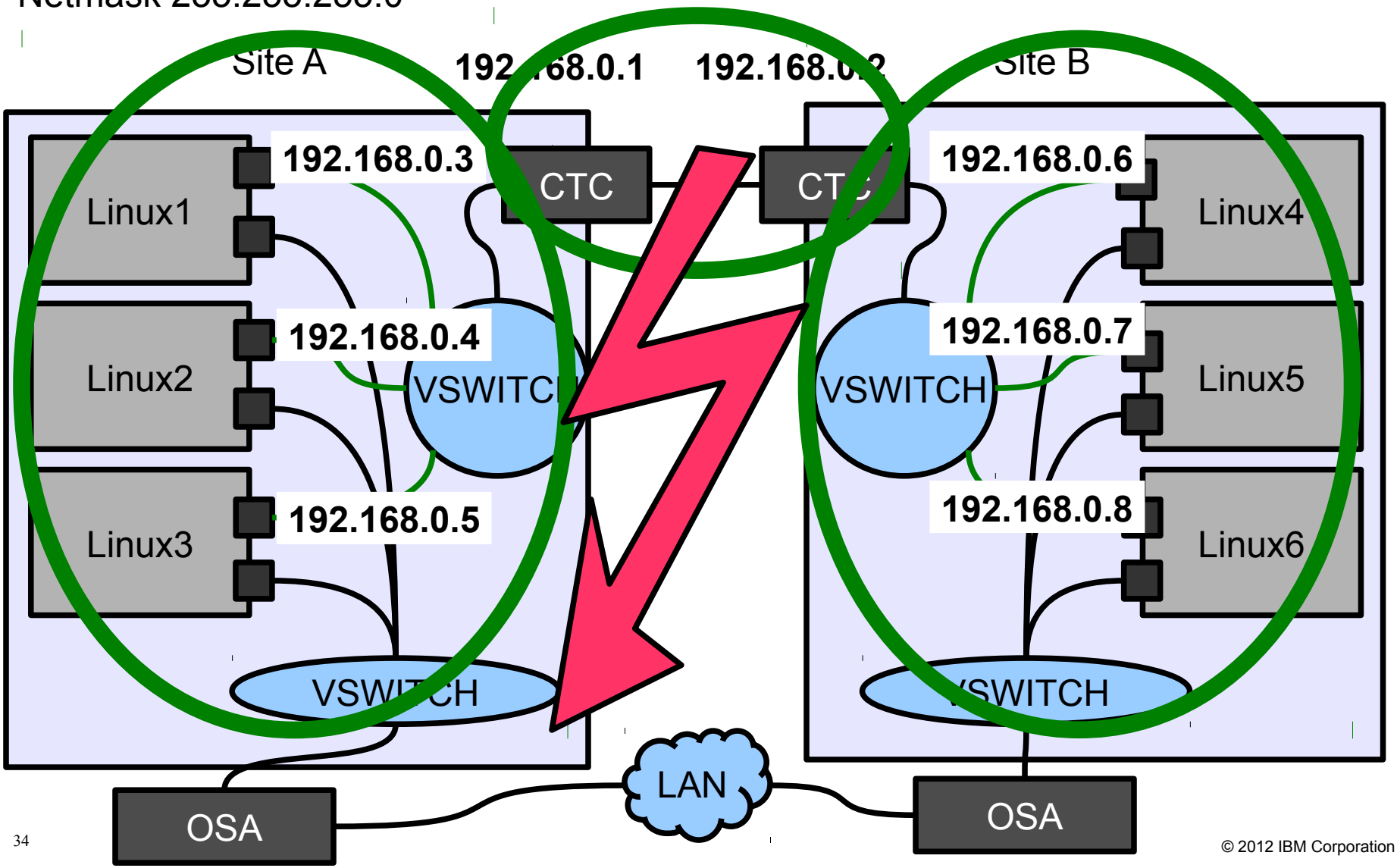


Where is the Problem?

Netmask 255.255.255.0

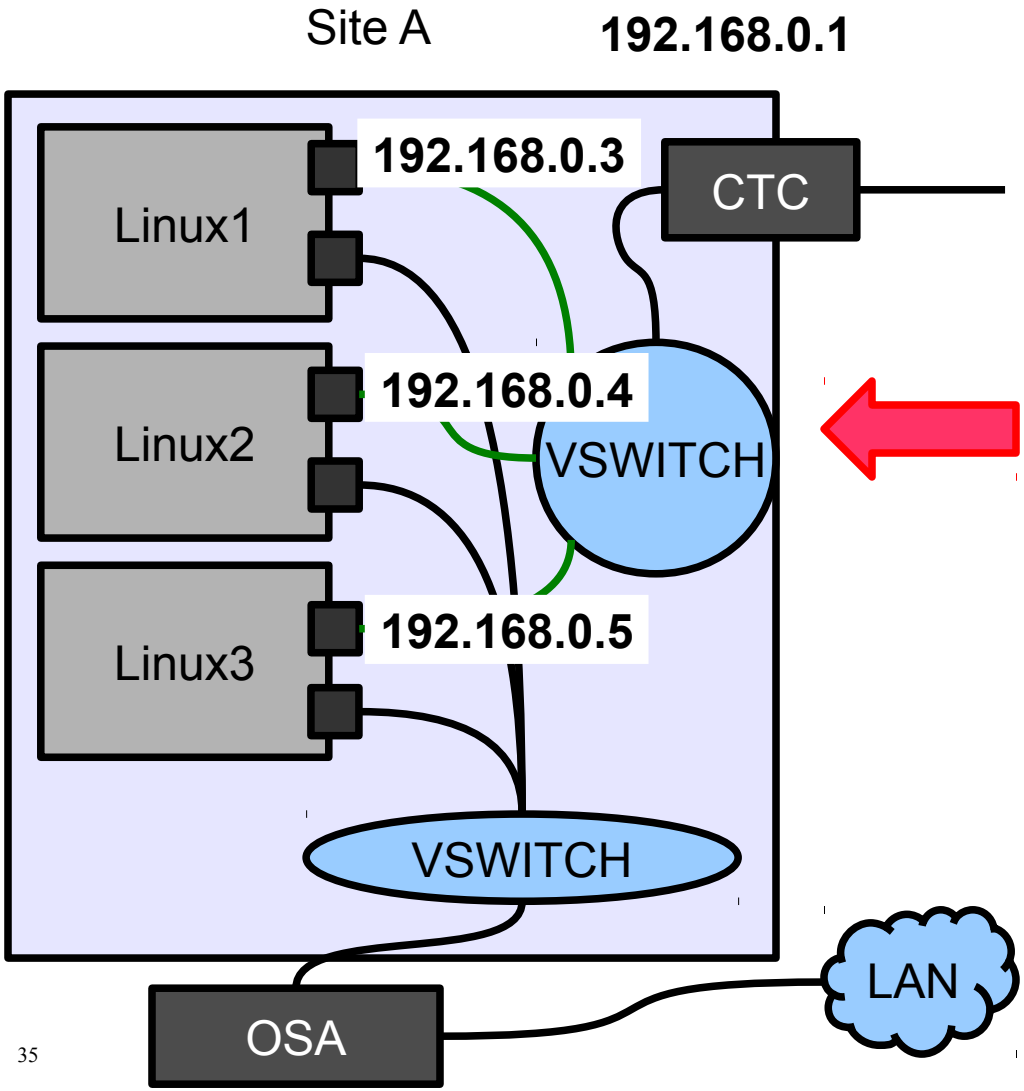


Where is the Problem?



Where is the Problem?

Netmask 255.255.255.0

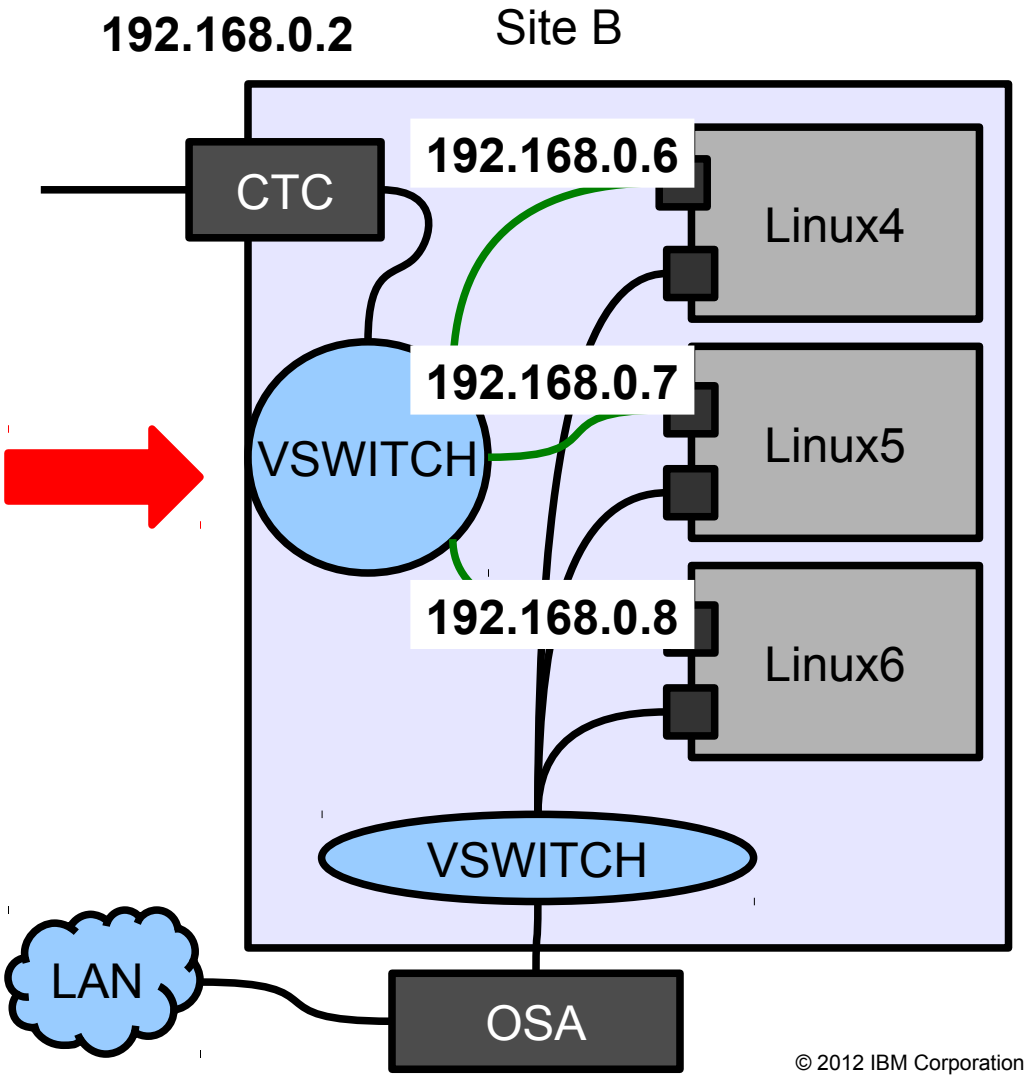


I am responsible for 192.168.0.0/24

Where is the Problem?

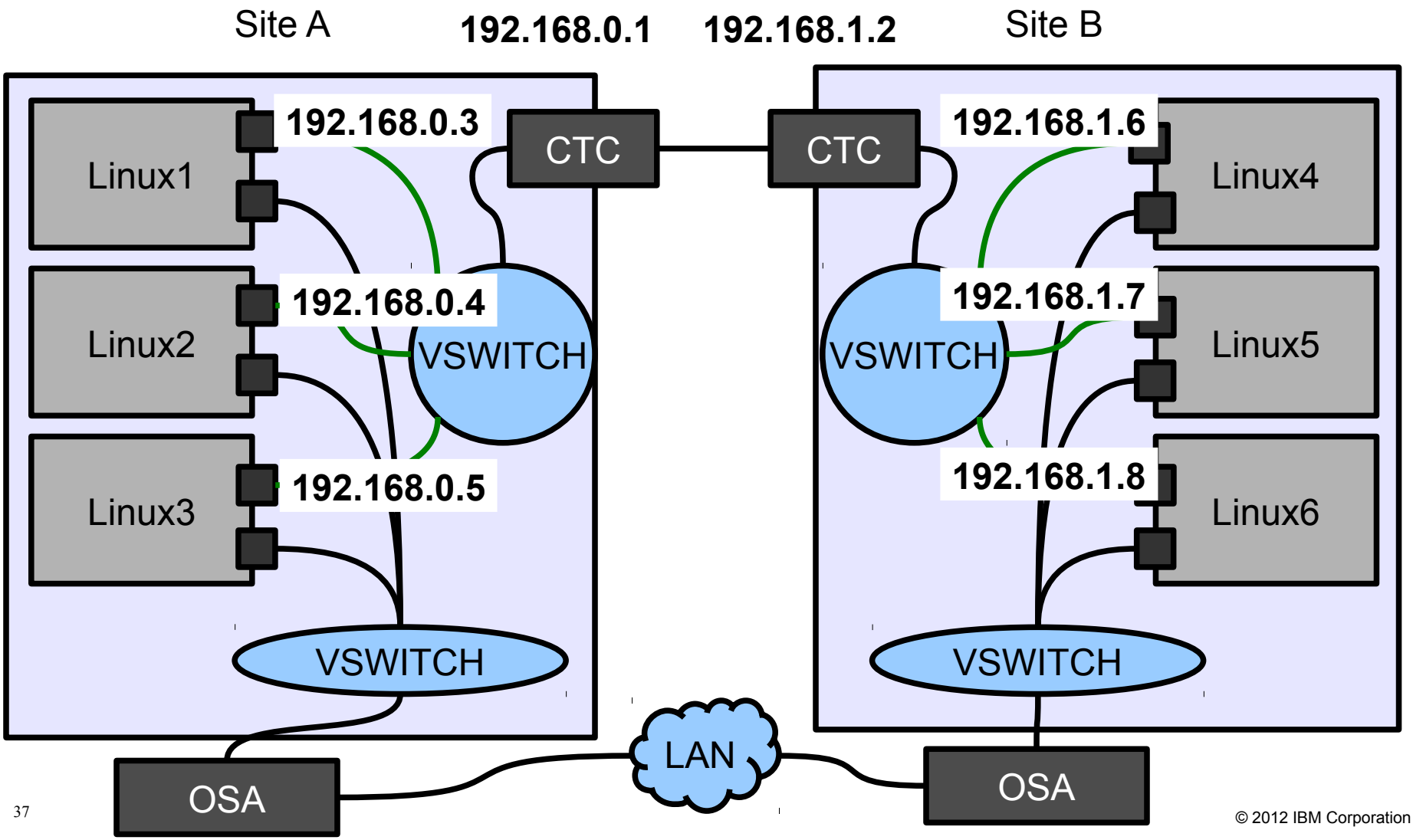
Netmask 255.255.255.0

I am responsible for 192.168.0.0/24



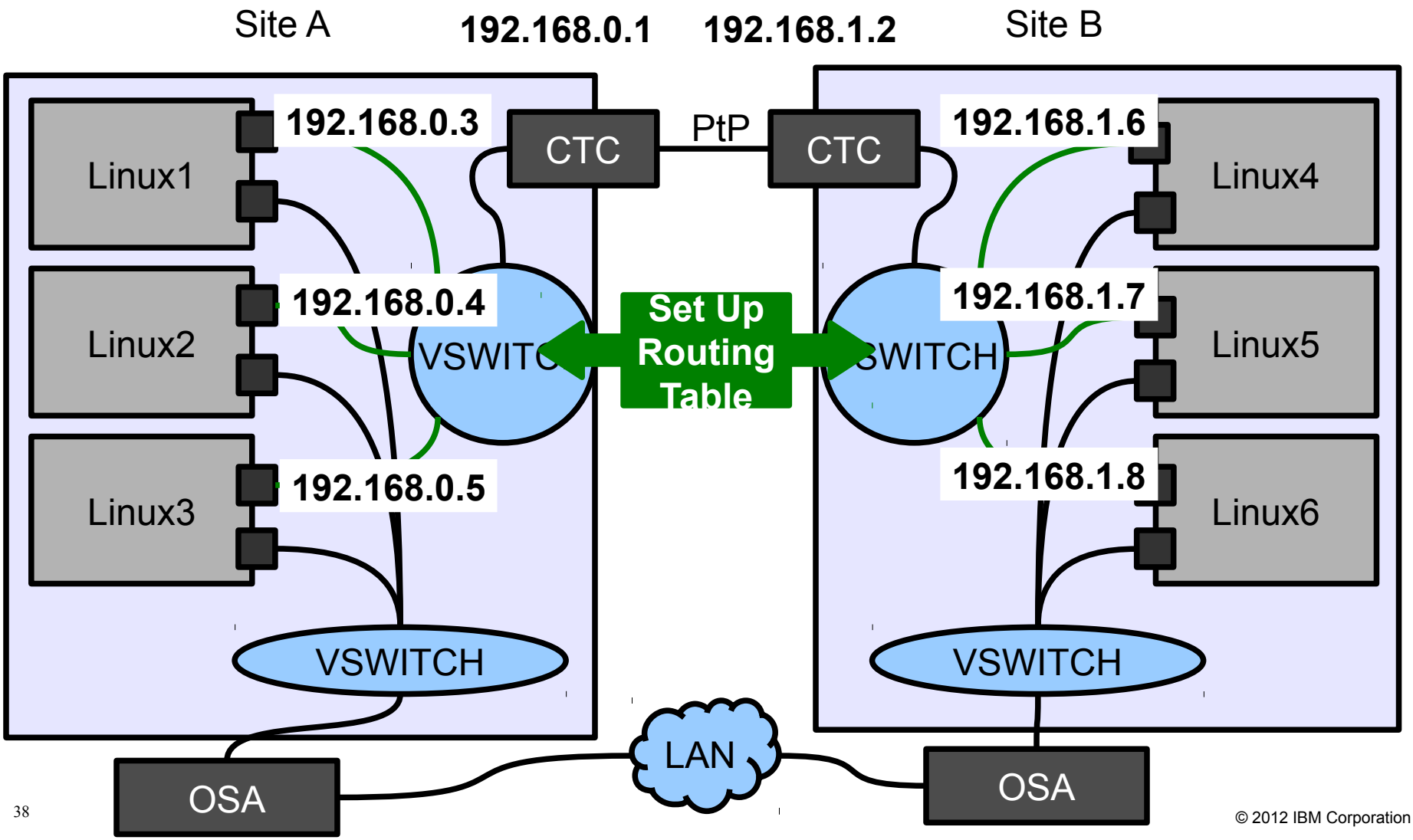
Where is the Problem?

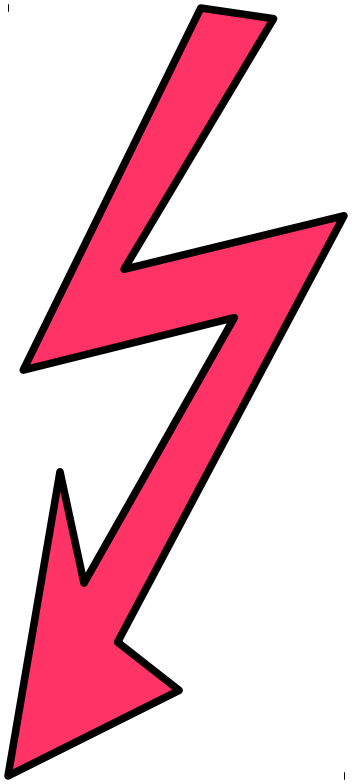
Netmask 255.255.255.0



Where is the Problem?

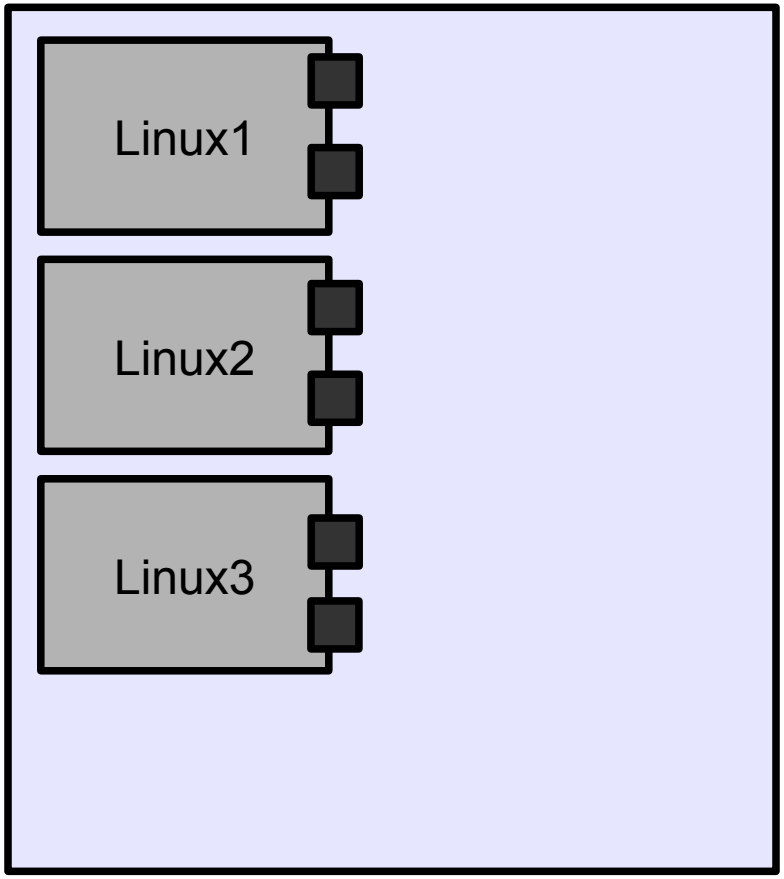
Netmask 255.255.255.0





This was not a System z specific problem!

Where is the Problem?



The customer has Linux Systems with multiple network interfaces!

Where is the Problem?

root@Linux1:~/ > **route -n**

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
10.9.15.0	0.0.0.0	255.255.255.0	U	0	0	0	hsi0
10.9.14.0	0.0.0.0	255.255.255.0	U	0	0	0	eth0
10.9.16.0	0.0.0.0	255.255.255.0	U	0	0	0	eth3
169.254.0.0	0.0.0.0	255.255.0.0	U	1002	0	0	hsi0
169.254.0.0	0.0.0.0	255.255.0.0	U	1003	0	0	eth0
169.254.0.0	0.0.0.0	255.255.0.0	U	1004	0	0	eth3
0.0.0.0	10.9.14.2	0.0.0.0	UG	0	0	0	eth0

Ip of eth0 = 10.9.14.100

ip of eth3 = 10.9.16.100

hsi0 is not used at the moment

Where is the Problem?

My laptop ip is 10.12.32.198
netmask 255.255.255.0

I can ping 10.9.14.100 but not 10.9.16.100

Where is the Problem?

Here's is more information you might what to look at:

When I do a traceroute from my linux laptop to the 10.9.14.100 ip I get through:

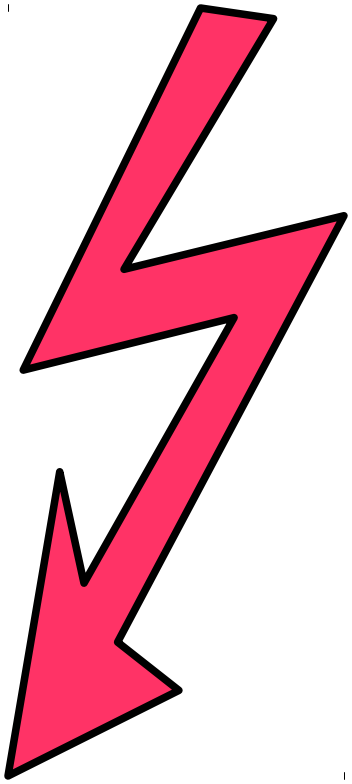
```
root@laptop:~/ > traceroute linux1
traceroute to linux1 (10.9.14.100), 30 hops max, 60 byte packets
1  10.12.32.3 (10.12.32.3)  0.763 ms  2.671 ms  2.682 ms
2  10.12.35.134 (10.12.35.134)  2.614 ms  2.547 ms  2.509 ms
3  X.X.X.X (X.X.X.X)  6.388 ms X.X.X.X (X.X.X.X)  6.367 ms X.X.X.X
(X.X.X.X)  6.337 ms
4  X.X.X.X (X.X.X.X)  10.233 ms  10.207 ms  10.187 ms
5  X.X.X.X (X.X.X.X)  10.124 ms  11.027 ms  9.972 ms
6  10.50.9.239 (10.50.9.239)  11.846 ms  11.745 ms  8.886 ms
7  10.9.1.6 (10.59.1.6)  8.856 ms  8.832 ms  8.799 ms
8  linux1 (10.9.14.100)  9.681 ms  9.654 ms  9.619 ms
```

Where is the Problem?

When I do the same to the the other IP (10.9.16.100) I do not get through

```
root@laptop:~/ > traceroute 10.9.16.100

raceroute to 10.9.16.100 (10.9.16.100), 30 hops max, 60 byte packets
1  10.12.32.3 (10.12.32.3)  1.496 ms  2.377 ms  2.370 ms
2  10.12.35.134 (10.12.35.134)  2.335 ms  4.229 ms  4.218 ms
3  X.X.X.X (X.X.X.X)  10.132 ms X.X.X.X (X.X.X.X)  7.086 ms 165.149.24.105
(1X.X.X.X)  10.051 ms
4  X.X.X.X (X.X.X.X)  10.910 ms  10.916 ms  10.840 ms
5  X.X.X.X (X.X.X.X)  9.785 ms  10.780 ms  10.707 ms
6  10.50.9.239 (10.50.9.239)  11.573 ms  8.608 ms  8.800 ms
7  10.9.1.14 (10.9.1.14)  8.775 ms  8.754 ms  8.718 ms
8  * * *
9  * * *
10  * * *
11  * * *
12  * * *
```

Where is the Problem?



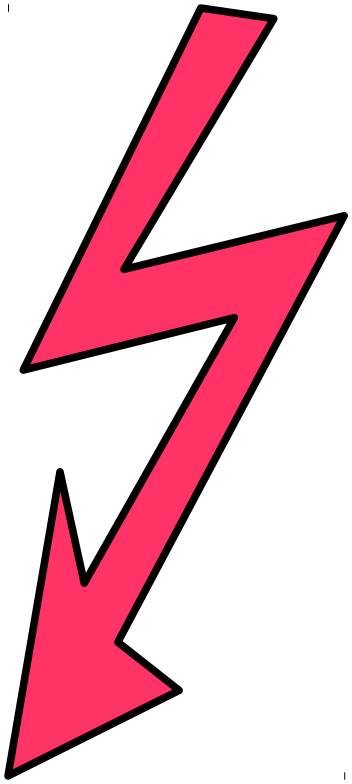
Where is the Problem?

On some proprietary Unix systems you can declare a default route per interface/network.

Linux instead has a "last default route declared is the default route"design.

This is why you experience here also a different behaviour between Linux and z/OS.

While Linux may, in fact, receive your connection request at the 10.9.16.100 address, its default routing behavior is to send it back out through its 10.9.14.100 address.



This was not a System z specific problem!

Solution: Linux's "advanced routing" functionality

```
ip route add 10.9.14.0/24 dev eth0 src 10.9.14.100 table 1
ip route add default via 10.9.14.2 dev eth0 table 1
ip rule add from 10.9.14.100/32 table 1
ip rule add to 10.9.14.100/32 table 1
ip route add 10.9.16.0/24 dev eth3 src 10.9.16.100 table 2
ip route add default via 10.9.16.2 dev eth3 table 2
ip rule add from 10.9.16.100/32 table 2
ip rule add to 10.9.16.100/32 table 2
```

The Linux Device Driver

Linux for System z Network Device Drivers

- QETH
- LCS
- CTC(M) (stabilized)
- NETIUCV (stabilized)

LAN Channel Station (LCS) Device Driver

- Supports:
  - OSA Express (in non-QDIO mode)
    - (HighSpeed TokenRing)
    - (ATM (running Ethernet LAN Emulation))
- May be preferred instead of QETH for security reasons
  - Administrator defines OSA Address Table → restricted access, whereas with QETH each Linux registers its own IP address
- But: performance is inferior to QETH's performance!!!

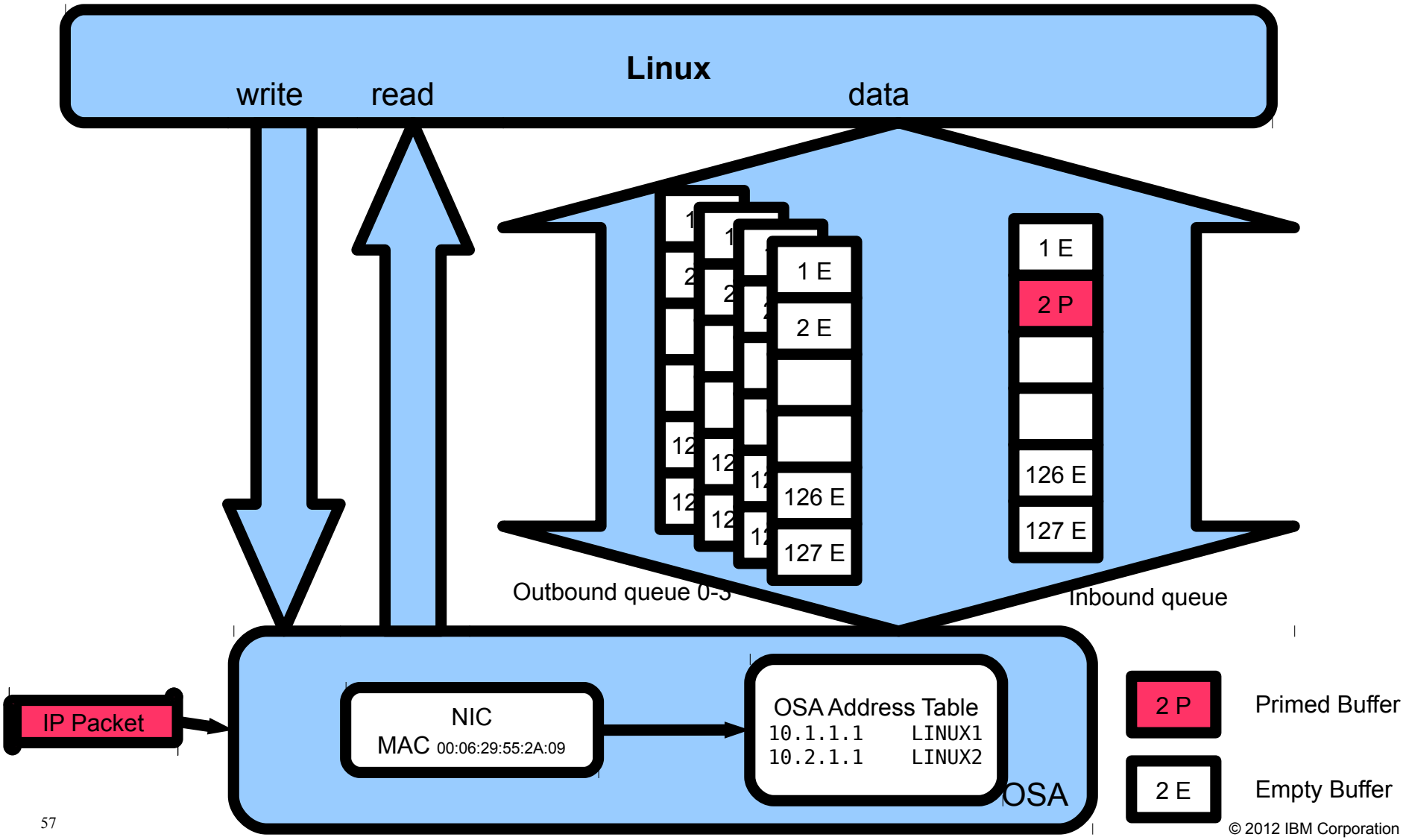
Message to CTC and IUCV users

- CTC = Channel-to-Channel connection
- IUCV = Inter User Communication Vehicle
- CTC(M) and NETIUCV device drivers are deprecated (Linux 2.6+)
- Device drivers are still available for backward compatibility
- Please consider migration
  - Virtual CTC and IUCV (under z/VM) ==> guest LAN HiperSocket or guest LAN type QDIO
  - CTC inside a CEC ==> Hipersockets
  - CTC ==> OSA-Express (QDIO)

QETH Device Driver

- Supports
  - OSA Express / OSA Express2 / OSA Express3 – OSD type (=QDIO)
    - Fast/Giga/10GBit Ethernet (fiber infrastructure)
    - 1000Base-T Ethernet (copper infrastructure)
  - System z HiperSockets
  - z/VM
    - GuestLAN Type QDIO (layer2 / layer3), Type Hiper
    - z/VM VSWITCH (layer2 / layer3)
  - IPv4, IPv6, VLAN, VIPA, Proxy ARP, IP Address Takeover, Channel Bonding
- Primary network driver for Linux on System z
- Main focus in current and future development

Queued Direct I/O (QDIO) Architecture

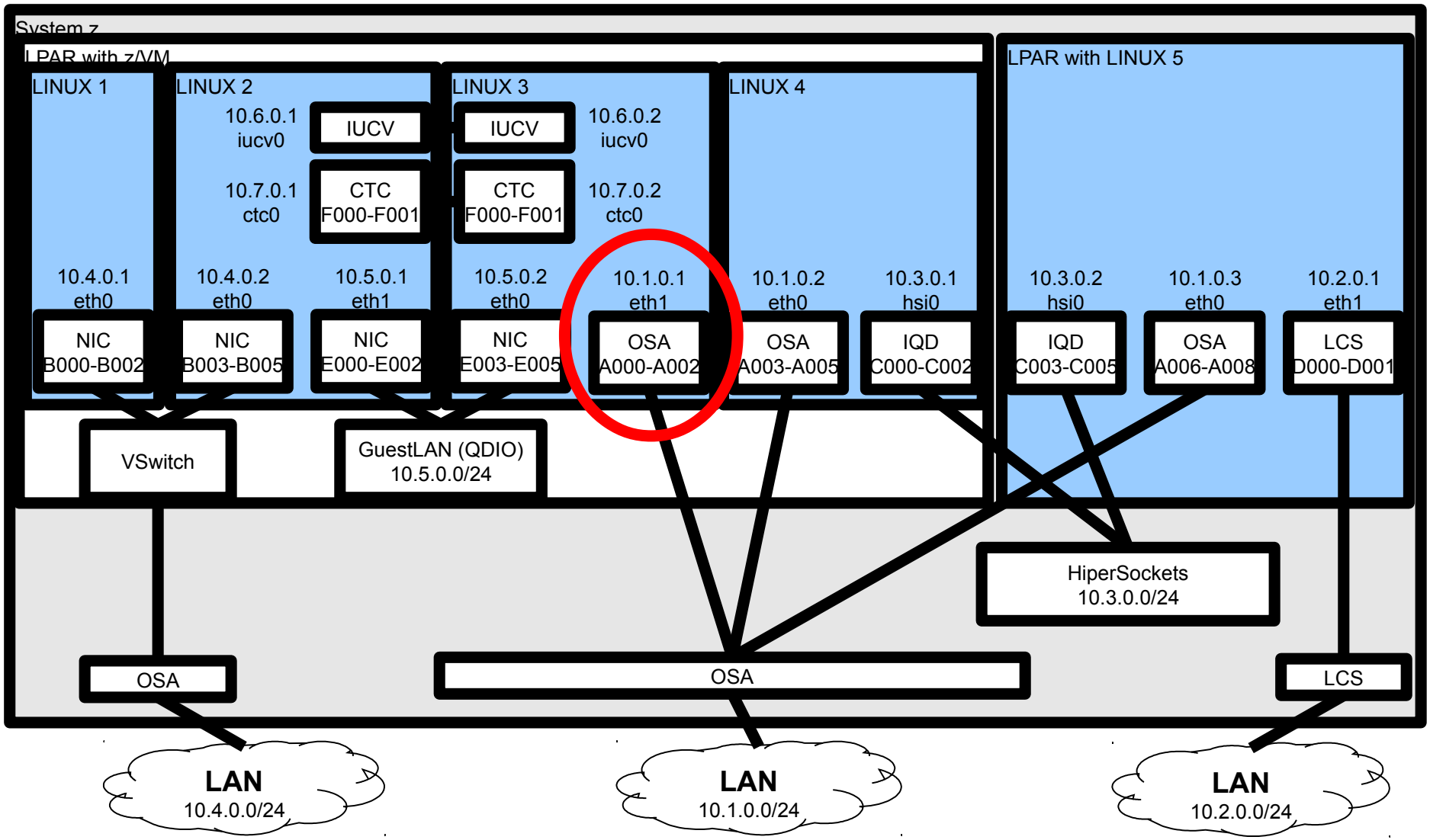


Network Configuration

Network Device Configuration (QETH)

- Example Network
- Generic (manual)
- Novell/SuSE SLES 10 / 11
- Red Hat RHEL 5 / 6

Networking Device Configuration – Example



Network Device Configuration – Generic

- Load the device driver module

```
root@larsson:~> modprobe qeth
```

- Create a new device by grouping its CCW devices:

```
root@larsson:~> echo 0.0.a000,0.0.a001,0.0.a002  > \
/sys/bus/ccwgroup/drivers/qeth/group
```

- Set optional attributes:

```
root@larsson:~> echo 64 > /sys/devices/qeth/0.0.a000/buffer_count
```

- Set the device online:

```
root@larsson:~> echo 1 > /sys/devices/qeth/0.0.a000/online
```

  - automatically assigns an interface name to the qeth device:
    - eth[n] for OSA devices
    - hsi[n] for HiperSocket devices

- Configure an IP address:

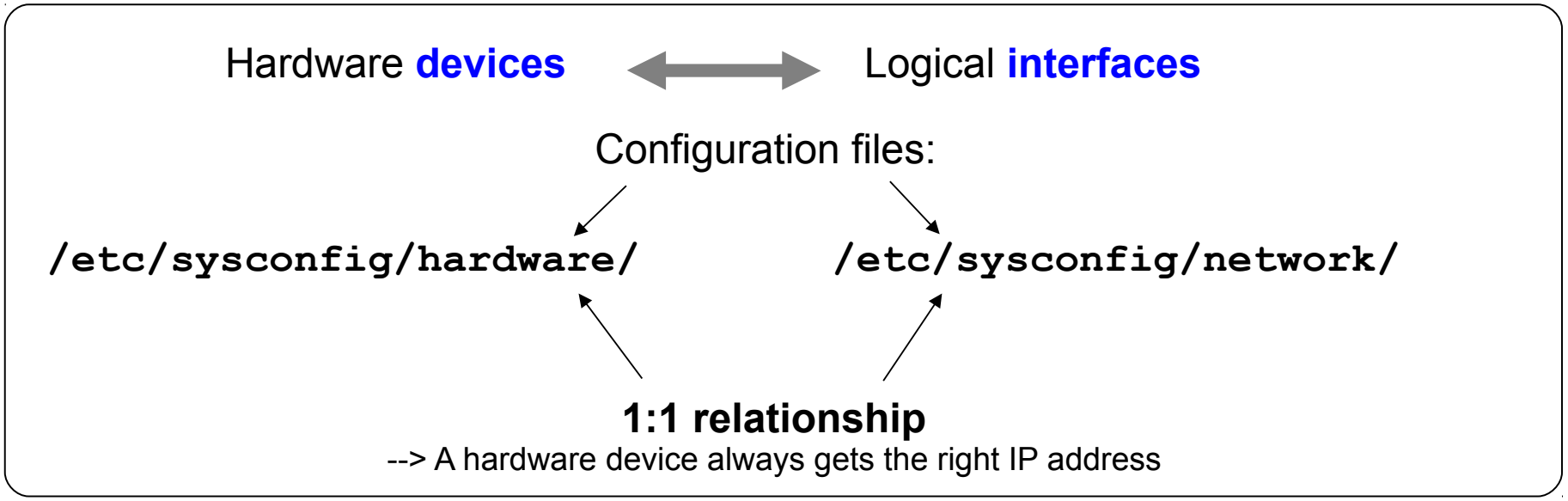
```
root@larsson:~> ifconfig eth0 10.1.0.1 netmask 255.255.255.0
```

The previous page is only important in case something goes wrong and your network connection does not come up automatically!

...unless you prefer sed & awk via 3270 over vi & emacs via ssh

You should use the distribution tools for the configuration....but it is always good to understand what is happening in the background

Network Device Configuration – SLES 10



Naming convention:

```
(hw|if)cfg-<device type>-bus-<bus type>-<bus location>
    e.g.   hwcfg-qeth-bus-ccw-0.0.a000
           ifcfg-qeth-bus-ccw-0.0.a000
```

Scripts:

```
hwup / hwdown, ifup / ifdown
```

see */etc/sysconfig/hardware/skel/hwcfg-<device type>*

Network Device Configuration – SLES 10 (cont'd)

1. Create an OSA hardware device configuration file:
   /etc/sysconfig/hardware/hwcfg-qeth-bus-ccw-0.0.a000

```
CCW_CHAN_IDS='0.0.a000 0.0.a001 0.0.a002'
CCW_CHAN_MODE='OSAPORT'
CCW_CHAN_NUM='3'
MODULE='qeth'
MODULE_OPTIONS=''
MODULE_UNLOAD='yes'
SCRIPTDOWN='hwdown-ccw'
SCRIPTUP='hwup-ccw'
SCRIPTUP_ccw='hwup-ccw'
SCRIPTUP_ccwgroup='hwup-qeth'
STARTMODE='auto'
QETH_LAYER2_SUPPORT='0'
QETH_OPTIONS='checksumming=hw_checksumming'
```

Network Device Configuration – SLES 10 (cont'd)

2. Create an interface configuration file
   /etc/sysconfig/network/ifcfg-qeth-bus-ccw-0.0.a000

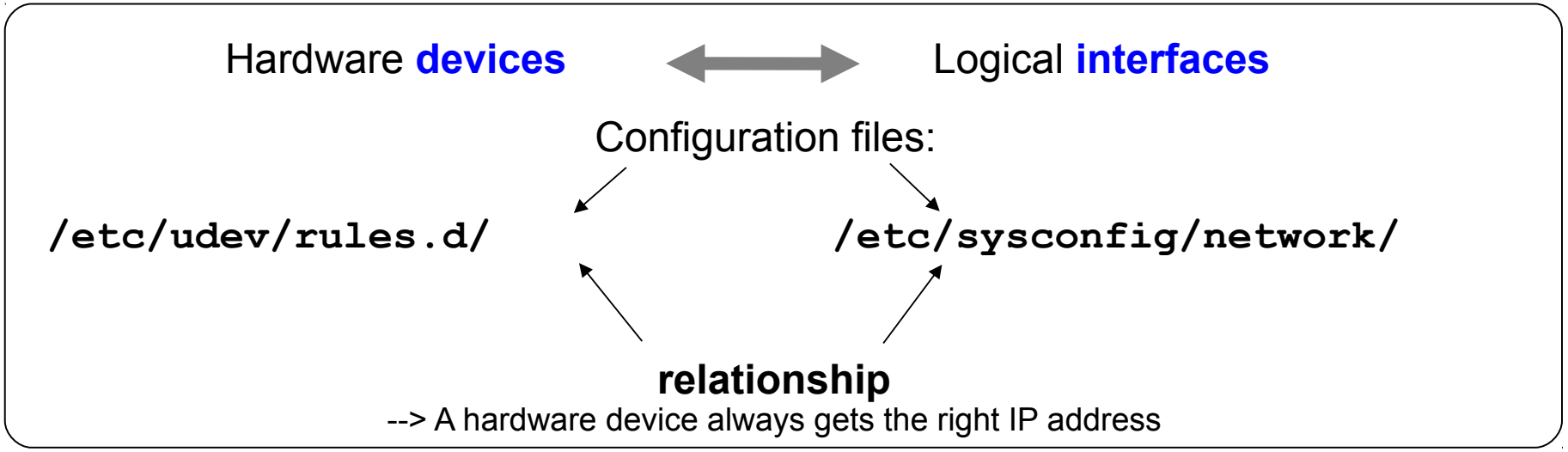
```
BOOTPROTO='static'
BROADCAST='10.1.0.255'
IPADDR='10.1.0.1'
NETMASK='255.255.255.0'
NETWORK='10.1.0.0'
STARTMODE='onboot'
```

For more details about configuration variables see:
/etc/sysconfig/network/ifcfg.template

3. Before you reboot: Test the configuration, bring the device up and then the interface

```
root@larsson:~> hwup qeth-bus-ccw-0.0.a000
root@larsson:~> ifup qeth-bus-ccw-0.0.a000
```

Network Device Configuration – SLES 11



Devices are configured via udev (framework for dynamic device configuration)

Naming convention:

```
51-<device type>-<bus location>.rules
70-persistent-net.rules
ifcfg-<interface name>
    e.g.   51-qeth-bus-0.0.a000.rules
           ifcfg-eth0
```

Scripts: `qeth_configure` and `ifup/ifdown`

Network Device Configuration – SLES 11 (cont'd)

1. "`qeth_configure -p OSAPORT 0.0.a000 0.0.a001 0.0.a002 1`" → */etc/udev/rules.d/51-qeth-bus-0.0.a000.rules*

```
# Configure qeth device at 0.0.a000/0.0.a001/0.0.a002
ACTION=="add", SUBSYSTEM=="drivers", KERNEL=="qeth",
IMPORT{program}="collect 0.0.a000 %k 0.0.a000 0.0.a001 0.0.a002 qeth"
ACTION=="add", SUBSYSTEM=="ccw", KERNEL=="0.0.a000",
IMPORT{program}="collect 0.0.a000 %k 0.0.a000 0.0.a001 0.0.a002 qeth"
ACTION=="add", SUBSYSTEM=="ccw", KERNEL=="0.0.a001",
IMPORT{program}="collect 0.0.a000 %k 0.0.a000 0.0.a001 0.0.a002 qeth"
ACTION=="add", SUBSYSTEM=="ccw", KERNEL=="0.0.a002",
 IMPORT{program}="collect 0.0.a000 %k 0.0.a000 0.0.a001 0.0.a002 qeth"
TEST=="[ccwgroup/0.0.a000]", GOTO="qeth-0.0.a000-end"
ACTION=="add", SUBSYSTEM=="ccw", ENV{COLLECT_0.0.a000}=="0",
ATTR{[drivers/ccwgroup:qeth]group}="0.0.a000,0.0.a001,0.0.a002"
ACTION=="add", SUBSYSTEM=="drivers", KERNEL=="qeth", ENV{COLLECT_0.0.a000}=="0",
ATTR{[drivers/ccwgroup:qeth]group}="0.0.a000,0.0.a001,0.0.a002"
LABEL="qeth-0.0.a000-end"
ACTION=="add", SUBSYSTEM=="ccwgroup", KERNEL=="0.0.a000", ATTR{layer2}="0"
ACTION=="add", SUBSYSTEM=="ccwgroup", KERNEL=="0.0.a000", ATTR{portname}="OSAPORT"
ACTION=="add", SUBSYSTEM=="ccwgroup", KERNEL=="0.0.a000", ATTR{portno}="0"
ACTION=="add", SUBSYSTEM=="ccwgroup", KERNEL=="0.0.a000", ATTR{online}="1"
```

Has to be first

Network Device Configuration – SLES 11 (cont'd)

2. Mapping between hardware device and Linux device
   /etc/udev/rules.d/70-persistent-net.rules

```
...
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="qeth", \
 KERNELS=="0.0.a000",ATTR{type}=="1",KERNEL=="eth*", NAME="eth0"
...
```

3. Create an interface configuration file
   /etc/sysconfig/network/ifcfg-eth0 (For more details about configuration variables see: */etc/sysconfig/network/ifcfg.template*)

```
BOOTPROTO='static'
BROADCAST='10.1.0.255'
IPADDR='10.1.0.1'
NETMASK='255.255.255.0'
NETWORK='10.1.0.0'
STARTMODE='onboot'
```

4. Before you reboot: Test the configuration

```
root@larsson:~> udevadm trigger
root@larsson:~> ifup eth0
```

Network Device Configuration – RHEL 5

- Configuration files

  /etc/modprobe.conf

```
alias eth0 qeth
alias eth1 qeth
alias hsi0 qeth
alias eth2 lcs
```

  /etc/sysconfig/network-scripts/ifcfg-<ifname>

```
NETTYPE         qeth | lcs
TYPE            Ethernet
SUBCHANNELS     0.0.a000,0.0.a001,0.0.a002
PORTNAME
OPTIONS                 e.g.  “layer2=1,portno=1”
MACADDR
```

- **ifup/ifdown** scripts contain mainframe-specifics

Network Device Configuration – RHEL 5 (cont'd)

1. Create a network device configuration file:
   /etc/sysconfig/network-scripts/ifcfg-eth0

```
DEVICE=eth0
SUBCHANNELS='0.0.a000 0.0.a001 0.0.a002'
PORTNAME='OSAPORT'
NETTYPE='qeth'
TYPE='Ethernet'
BOOTPROTO=static
ONBOOT=yes
BROADCAST='10.1.0.255'
IPADDR='10.1.0.1'
NETMASK='255.255.255.0'
NETWORK='10.1.0.0‘
MACADDR='00:09:6B:1A:9A:89‘
OPTIONS='layer2=0'
```

Network Device Configuration – RHEL 5 (cont'd)

2. Add / verify alias in module configuration file */etc/modprobe.conf*

```
…
alias eth0 qeth
…
```

For further information, see

http://www.redhat.com/docs/manuals/enterprise

Network Device Configuration – RHEL 6

1. Create a network device configuration file:
   /etc/sysconfig/network-scripts/ifcfg-eth0

```
DEVICE=eth0
SUBCHANNELS='0.0.a000,0.0.a001,0.0.a002'
PORTNAME='OSAPORT'
NETTYPE='qeth'
TYPE='Ethernet'
BOOTPROTO=static
ONBOOT=yes
IPADDR='10.1.0.1'
NETMASK='255.255.255.0'
NETWORK='10.1.0.0'
OPTIONS='layer2=0 portno=0'
```

Network Device Configuration – RHEL 6 (cont'd)

2. Mapping between hardware device and Linux device
   /etc/udev/rules.d/70-persistent-net.rules

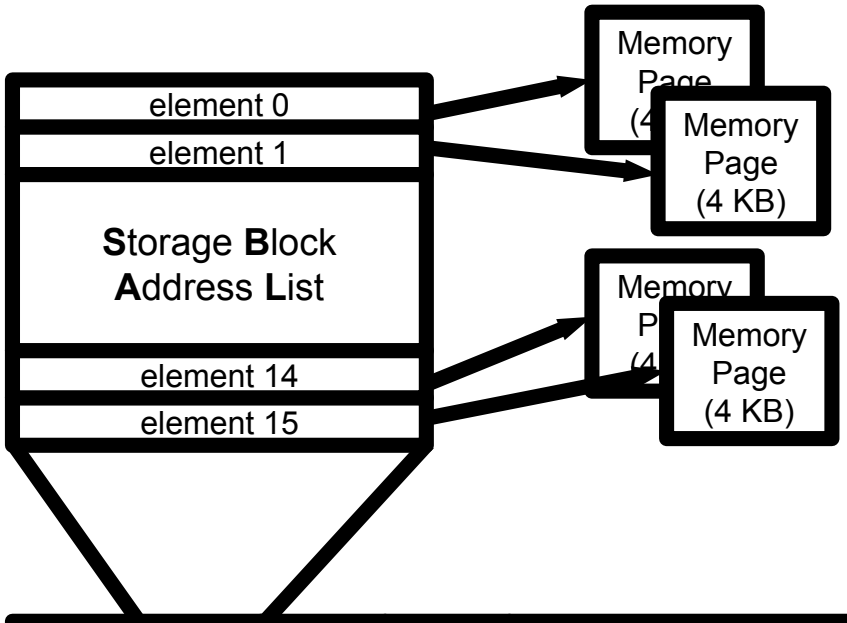
```
…
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*",
ENV{INTERFACE_NAME}=="eth0", DRIVERS=="qeth", KERNELS=="0.0.a000",
ATTR{type}=="1", KERNEL=="eth*", NAME="eth0"
…
```

3. Before you reboot: Test the configuration

```
root@larsson:~> ifup eth0
```

QETH Device sysfs Attribute buffer_count

- The number of allocated buffers for inbound QDIO traffic
  → Memory usage.



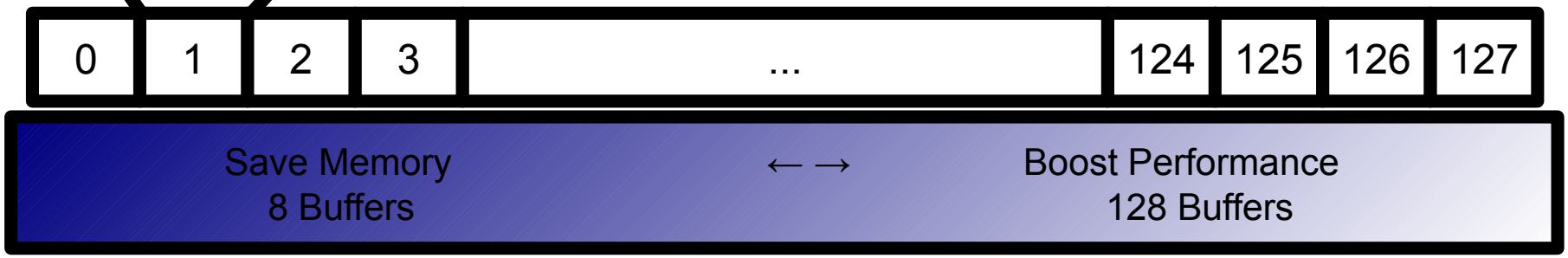
Per QETH device memory usage:

control data structures: ~ 200 KB
memory for one buffer: 64 KB

buffer_count = 8 --> ~ 712 KB

buffer_count = 128 --> ~ 8.4 MB

(hipersocket numbers depend on MTU size)



QETH Device sysfs Attribute checksumming

- Additional redundancy check to protect data integrity
- Offload checksumming for incoming IP packages from Linux stack to OSA-card

```
QETH_OPTIONS='checksumming=hw_checksumming'   or
echo hw_checksumming > …  or
ethtool -K …
```

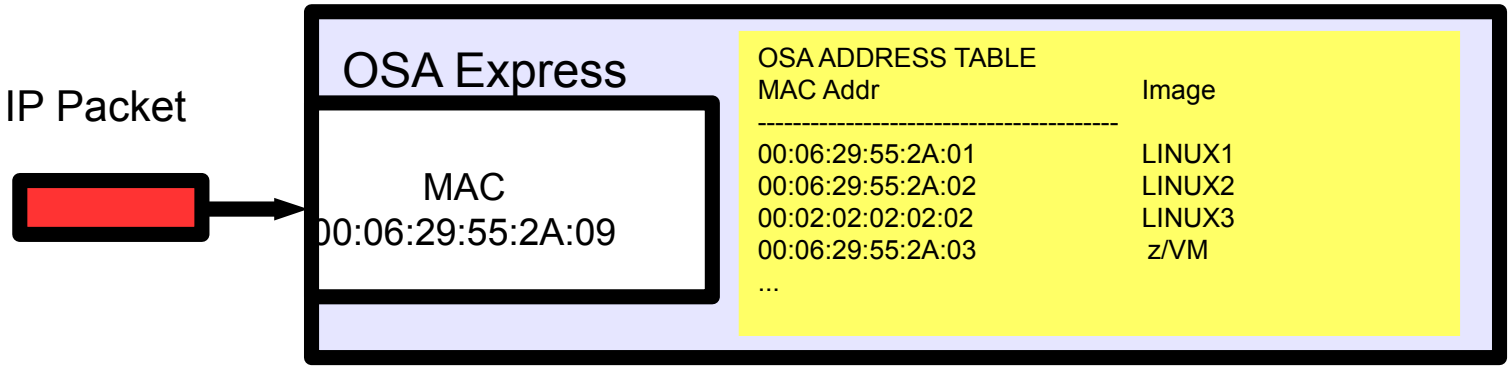
```
root@larsson:~> echo hw_checksumming > \
/sys/devices/qeth/0.0.a000/checksumming

root@larsson:~> ethtool -K eth0 rx on
```

===> move workload from Linux to OSA-Express adapter

- Available for OSA-devices in layer3 mode only

QETH Layer 2 mode

- OSA works with MAC address ==> no longer stripped from packets



hwcfg-qeth... file (SLES10): QETH_LAYER2_SUPPORT=1

ifcfg-qeth... file (SLES10): LLADDR='<MAC Address>'

51.qeth....rule (SLES11): ATTR{layer2}="1"

ifcfg-qeth... file (SLES11): LLADDR='<MAC Address>'

ifcfg-... file (RHEL5/6): MACADDR='<MAC Address>'

OPTIONS='layer2=1'

QETH Layer 2 mode (cont'd)

- Direct attached OSA:
  - MAC address must be defined manually with ifconfig
    ```
    ifconfig eth0 hw ether 00:06:29:55:2A:01
    ```
  - Restrictions: Older OSA-generation (≤ z990):
    Layer2 and Layer3 traffic can be transmitted over the same OSA CHPID, but not between two images sharing the same CHPID !
- Hipersockets:
  - new layer2 support starting with z10 - MAC address automatically generated
  - Layer2 and Layer3 traffic separated
- VSWITCH or GuestLAN under z/VM: MAC address created by z/VM

```
define lan <lanname> ... type QDIO ETHERNET
      define nic <vdev> QDIO
      couple <vdev> <ownerid> <lanname>
define vswitch <vswname> ... ETHERNET …
      define nic <vdev> QDIO
      couple <vdev> <ownerid> <lanname>
```

QETH Layer 2 mode (cont'd)

- activating Layer 2 is done
  per device via sysfs attributes
- possible layer2 values:
  - 0: use device in Layer 3 mode
  - 1: use device in Layer 2 mode

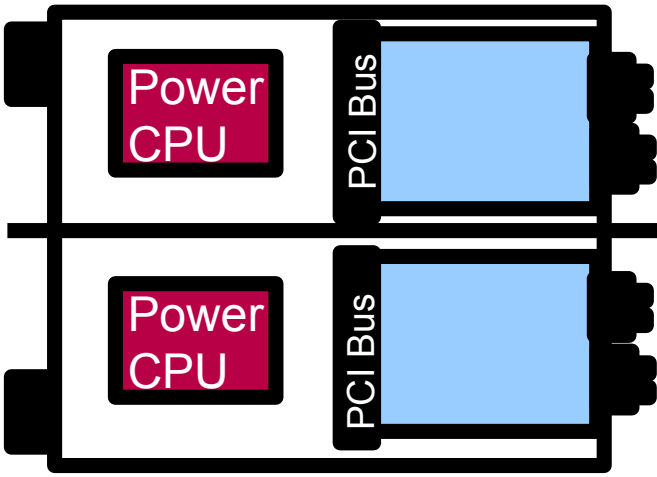
```
/sys
|--devices
   |--qeth
      |--0.0.<devno>
         |--layer2
```

- setting of layer2 attribute only permitted when device is offline!
- Advantages:
  - Independent of IP-protocol or any layer3 protocol
  - channel bonding possible

OSA Express 3 – 2 ports within 1 chpid

- OSA Express2 – 2 CHPIDs with 1 port per CHPID – 2 ports totally
- OSA Express3 – 2 CHPIDs with 2 ports per CHPID – 4 ports totally (>= z10)
- New sysfs-attribute “portno” can contain '0' or '1'
- OSA-Express3 GbE SX and LX

hwcfg-qeth... file (SLES10):	QETH_OPTIONS=”portno=1”
51.qeth....rule (SLES11):	ATTR{portno}="1"
Or command (SLES11):	`qeth_configure -n 1` ....
ifcfg-... file (RHEL5/6):	OPTIONS='portno=1



Commands / tools for qeth-driven devices

- List of known qeth devices: cat /proc/qeth or lsqeth -p

```
root@larsson:~> cat /proc/qeth
devices                  CHPID interface cardtype   port chksum
 ------------------------ ----- --------- ---------  ---- ------
0.0.a000/0.0.a001/0.0.a002 xA0   eth0      OSD_1000     0    sw
0.0.c000/0.0.c001/0.0.c002 xC0   hsi0      HiperSockets 0    sw
```

- Attributes of qeth device: lsqeth or lsqeth <interface>

```
root@larsson:~> lsqeth eth0
Device name                : eth0
-----------------------------------------
        card_type            : OSD_1000
        cdev0               : 0.0.a000
        cdev1               : 0.0.a001
        cdev2               : 0.0.a002
        chpid               : 76
        online              : 1
        state               : UP (LAN ONLINE)
        buffer_count           : 16
        layer2              : 0
```

Commands / tools for qeth-driven devices: znetconf

- Allows the user to list, add, remove & configure System z network devices
- To list all configured network devices:

```
root@larsson:~>  znetconf -c
Device IDs                Type    Card Type CHPID Drv. Name  State
------------------------------------------------------------------
0.0.a000,0.0.a001,0.0.a002 1731/01 OSD_1000     76 qeth eth0  online
```

- To list all potential network devices, us the **-u** option
- Configure device 0.0.a000 in layer2 mode and with portnumber “1”

```
root@larsson:~>  znetconf -a a000 -o layer2=0 -n portno=1
```

- Remove network device 0.0.a000

```
root@larsson:~>  znetconf -r a000
```

Commands / tools for qeth-driven devices: ethtool

- Use ethtool to query, set and change attributes
- To query ethernet driver information:

```
root@larsson:~>  ethtool -i eth0
driver: qeth_l3
version: 1.0
firmware-version: 0893
bus-info: 0.0.a000/0.0.a001/0.0.a002
```

- To query the offload information of the specified ethernet device

```
root@larsson:~>  ethtool -k eth0
Offload parameters for eth0:
rx-checksumming: off
tx-checksumming: off .....
```

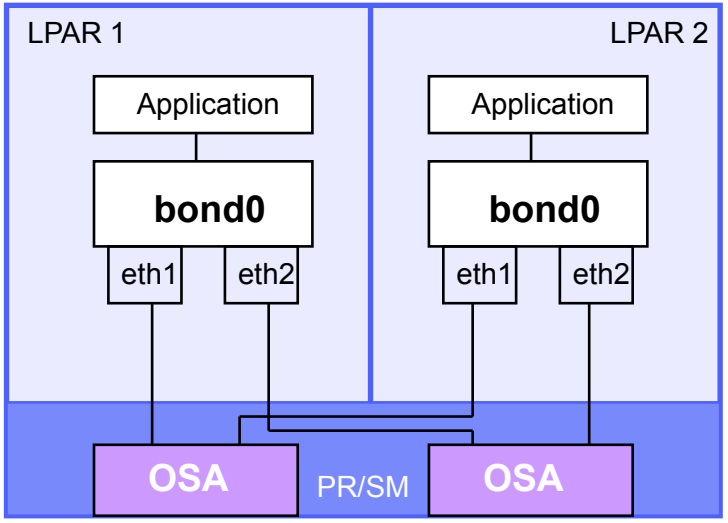
- Example: to change the inbound checksumming offload parameter

```
root@larsson:~>  ethtool -K eth0 rx on
```

Advanced Topics

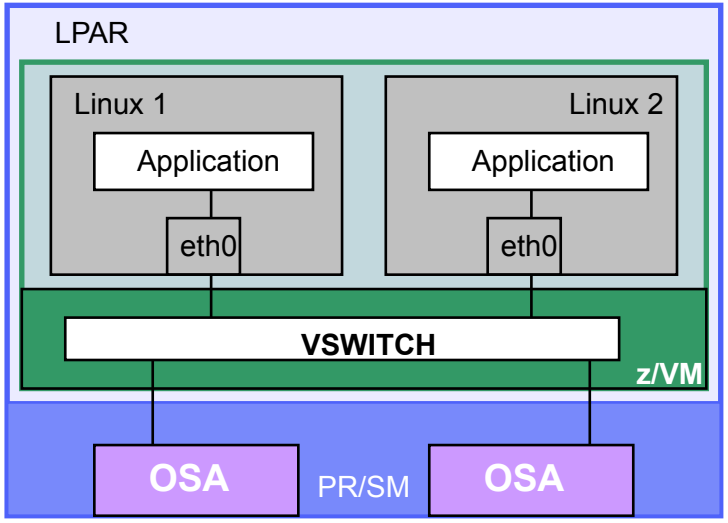
Leveraging Virtualization for Network Interface Redundancy and Automated Failover

Resource Virtualization: OSA Channel Bonding



- Linux *bonding* driver enslaves multiple OSA connections to create a single logical network interface card (NIC)
- Detects loss of NIC connectivity and automatically fails over to surviving NIC
- No dynamic routing (OSPF) dependency
- Active/backup & aggregation modes
- **Separately configured for each Linux**

System Virtualization: z/VM VSWITCH



- z/VM *VSWITCH* enslaves multiple OSA connections. Creates virtual NICs for each Linux guest
- Detects loss of physical NIC connectivity and automatically fails over to surviving NIC
- No dynamic routing (OSPF) dependency
- Active/backup & aggregation modes
- **Centralized configuration benefits all guests**

Channel Bonding Support

- The Linux bonding driver provides a method for aggregating multiple network interfaces into a single, logical “bonded” interface
- Provides failover and/or load-balancing functionality
- Better performance depending on bonding mode
- Applies to layer2-devices only
- Further information http://sourceforge.net/projects/bonding

Setting Up Channel Bonding

- Load bonding module with miimon option

```
root@larsson:~> modprobe bonding miimon=100 mode=balance-rr
```

(miimon option enables link monitoring)

- Add MAC addresses to slave devices eth0 & eth1 (not necessary for GuestLAN or Vswitch)

```
root@larsson:~> ifconfig eth0 hw ether 00:06:29:55:2A:01
root@larsson:~> ifconfig eth1 hw ether 00:05:27:54:21:04
```

- Activate the bonding device bond0

```
root@larsson:~> ifconfig bond0 10.1.1.1 netmask 255.255.255.0
```

- Connect slave devices eth0 & eth1 to bonding device bond0

```
root@larsson:~> ifenslave bond0 eth0 eth1
```

Setting Up Channel Bonding (SLES10/11)

- Interface configuration file for a slave device
  - SLES10: */etc/sysconfig/network/ifcfg-qeth-bus-ccw-0.0.a000*
  - SLES11: */etc/sysconfig/network/ifcfg-eth0*

```
BOOTPROTO='static'
IPADDR=''
SLAVE='yes'
STARTMODE='onboot'
LLADDR='00:06:29:55:2A:01'
```

Setting Up Channel Bonding (SLES10/11) (cont'd)

- Interface configuration file for a master device
  /etc/sysconfig/network/ifcfg-bond0

```
BOOTPROTO='static'
BROADCAST='10.1.1.255'
IPADDR='10.1.1.1'
NETMASK='255.255.255.0'
NETWORK='10.1.1.0'
STARTMODE='onboot'
BONDING_MASTER='yes'
BONDING_MODULE_OPTS='mode=active_backup fail_over_mac=active
      miimon=100'
LLADDR='00:06:29:55:2A:03'
# SLES10
BONDING_SLAVE0='qeth-bus-ccw-0.0.a000'
BONDING_SLAVE1='qeth-bus-ccw-0.0.b000'
# SLES11
BONDING_SLAVE0='eth0'
BONDING_SLAVE1='eth1'
```

Setting Up Channel Bonding (RHEL5/RHEL6)

- Interface configuration file for a slave device
  /etc/sysconfig/network/ifcfg-eth0

```
DEVICE=eth0
USERCTL='yes'
BOOTPROTO='none'
SLAVE='yes'
MASTER='bond0'
ONBOOT='yes'
SUBCHANNELS=0.0.a000,0.0.a001,0.0.a002
TYPE=Ethernet
OPTIONS="layer2=1"
MACADDR=00:06:29:55:2A:01
```

- Add / verify alias in module configuration file /etc/modprobe.conf (RHEL5 only)

```
…
alias bond0 bonding
options bond0 miimon=100 mode=active_backup fail_over_mac=active
…
```

Setting Up Channel Bonding (RHEL5/RHEL6) (cont'd)

- Interface configuration file for a master device
  /etc/sysconfig/network/ifcfg-bond0

```
DEVICE=bond0
BOOTPROTO='none'
BROADCAST='10.1.1.255'
IPADDR='10.1.1.1'
NETMASK='255.255.255.0'
NETWORK='10.1.1.0'
ONBOOT='yes'
USERCTL='yes'
NETTYPE='qeth'
TYPE=Bonding
MACADDR=00:06:29:55:2A:03
# RHEL6 only
BONDING_OPTS='mode=active_backup fail_over_mac=active miimon=100'
```

Setting Up Channel Bonding (cont'd)

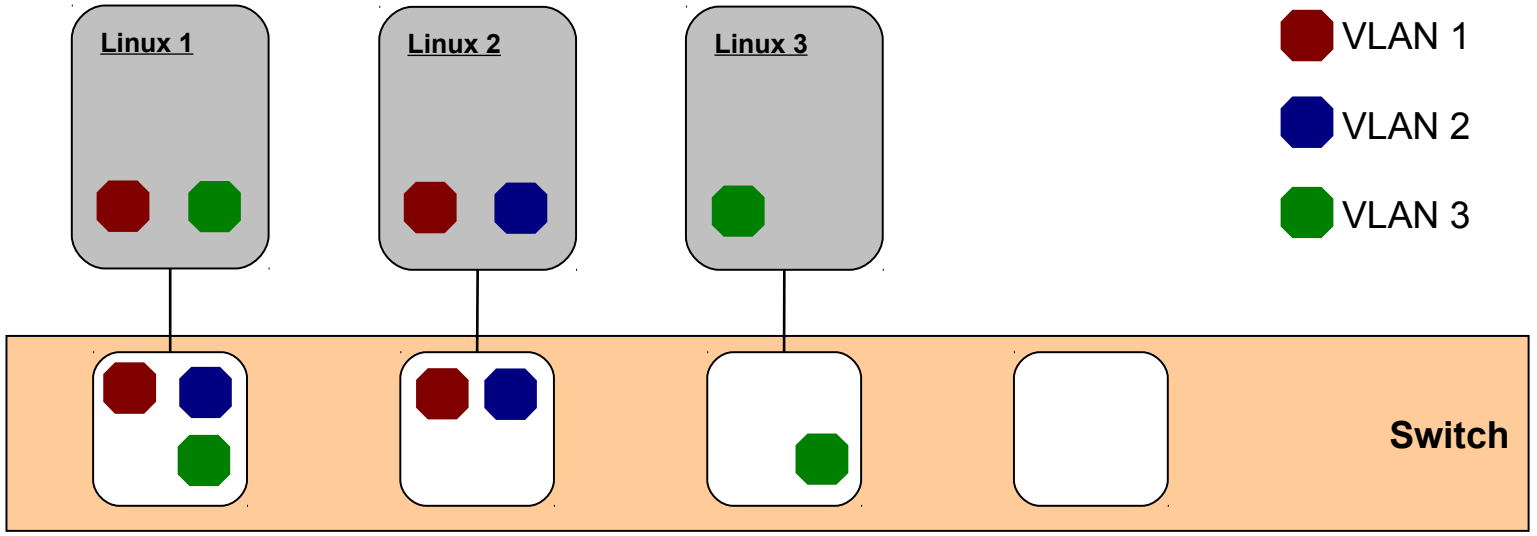
- Display the interface and bonding configuration

```
root@larsson:~> ifconfig
bond0     Link encap:Ethernet  HWaddr 00:06:29:55:2A:01
          inet addr:10.1.1.1  Bcast:10.255.255.255 ...

eth0      Link encap:Ethernet  HWaddr 00:06:29:55:2A:01
          UP BROADCAST RUNNING SLAVE MULTICAST  MTU:1500...

eth1      Link encap:Ethernet  HWaddr 00:06:29:55:2A:02
          UP BROADCAST RUNNING SLAVE MULTICAST  MTU:1500 ...
```

```
root@larsson:~> cat /proc/net/bonding/bond0
Bonding Mode: fault-tolerance (active-backup) (fail_over_mac active)
Primary Slave: None
Currently Active Slave: eth0
MII Status: up
MII Polling Interval (ms): 100
Slave Interface: eth0
MII Status: up
Permanent HW addr: 00:06:29:55:2A:01
Slave Interface: eth1
MII Status: up
Permanent HW addr: 00:06:29:55:2A:02
```

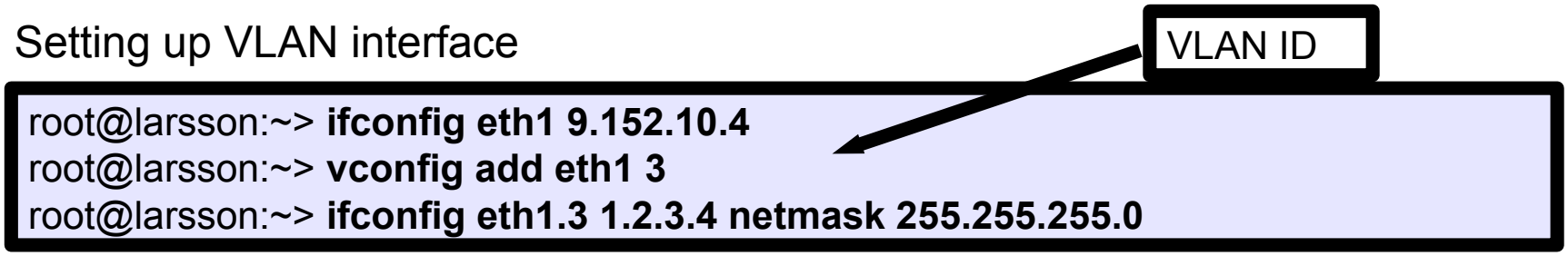
Virtual LAN (VLAN) Support

- IEEE Standard 802.1Q
- Reduce broadcast traffic
- Divide LANs logically into subnets to optimize bandwidth utilization
- Network devices supporting VLAN:
  - real OSA card, HiperSockets, z/VM GuestLAN, z/VM VSWITCH



Setting Up Virtual LAN (VLAN)

- Setting up VLAN interface



- Removing a VLAN interface

```
root@larsson:~> ifconfig eth1.3 down
root@larsson:~> vconfig rem eth1.3
```

- Displaying the VLAN configuration

```
root@larsson:~> cat /proc/net/vlan/config
VLAN Dev name    | VLAN ID
Name-Type: VLAN_NAME_TYPE_RAW_PLUS_VID_NO_PAD
eth1.3         | 3  | eth1
```

- Implementation:
  - VLAN tag added to transmitted frames and removed from received frames

Customer Cases

Network: network connection is too slow

Configuration:

- z/VSE running CICs, connection to DB2 in Linux on System z
- Hipersocket connection from Linux to z/VSE
- But also applies to hipersocket connections between Linux and z/OS

Problem Description:

- When CICS transactions were monitored, some transactions take a couple of seconds instead of milliseconds

Tools used for problem determination:

- dbginfo.sh
- s390 debug feature
- sadc/sar
- CICS transaction monitor

Network: network connection is too slow (cont'd)

- s390 debug feature
  - Check for qeth errors:

```
cat /sys/kernel/debug/s390dbf/qeth_qerr
00 01282632346:099575 2 - 00 0000000180b20218  71 6f 75 74 65 72 72 00 | qouterr.
00 01282632346:099575 2 - 00 0000000180b20298  20 46 31 35 3d 31 30 00 |  F15=10.
00 01282632346:099576 2 - 00 0000000180b20318  20 46 31 34 3d 30 30 00 |  F14=00.
00 01282632346:099576 2 - 00 0000000180b20390  20 71 65 72 72 3d 41 46 |  qerr=AF
00 01282632346:099576 2 - 00 0000000180b20408  20 73 65 72 72 3d 32 00 |  serr=2.
```

- dbginfo file
  - Check for buffer count:

```
cat /sys/devices/qeth/0.0.1e00/buffer_count
16
```

- Problem Origin:
  - Too few buffers

Network: network connection is too slow (cont'd)

Solution:

- Increase buffer count (default: 16, max 128)
- Check actual buffer count with `'lsqeth -p'`
- Set the buffer count in the appropriate config file:
  - SUSE SLES10:
    - in /etc/sysconfig/hardware/hwcfg-qeth-bus-ccw-0.0.F200
    - add `QETH_OPTIONS="buffer_count=128"`
  - SUSE SLES11:
    - in /etc/udev/rules.d/51-qeth-0.0.f200.rules add
      ACTION=="add",
      `SUBSYSTEM=="ccwgroup", KERNEL=="0.0.f200",`
      `ATTR{buffer_count}="128"`
  - Red Hat:
    - in /etc/sysconfig/network-scripts/ifcfg-eth0
    - add OPTIONS=`"buffer_count=128"`

Bonding throughput not matching expectations

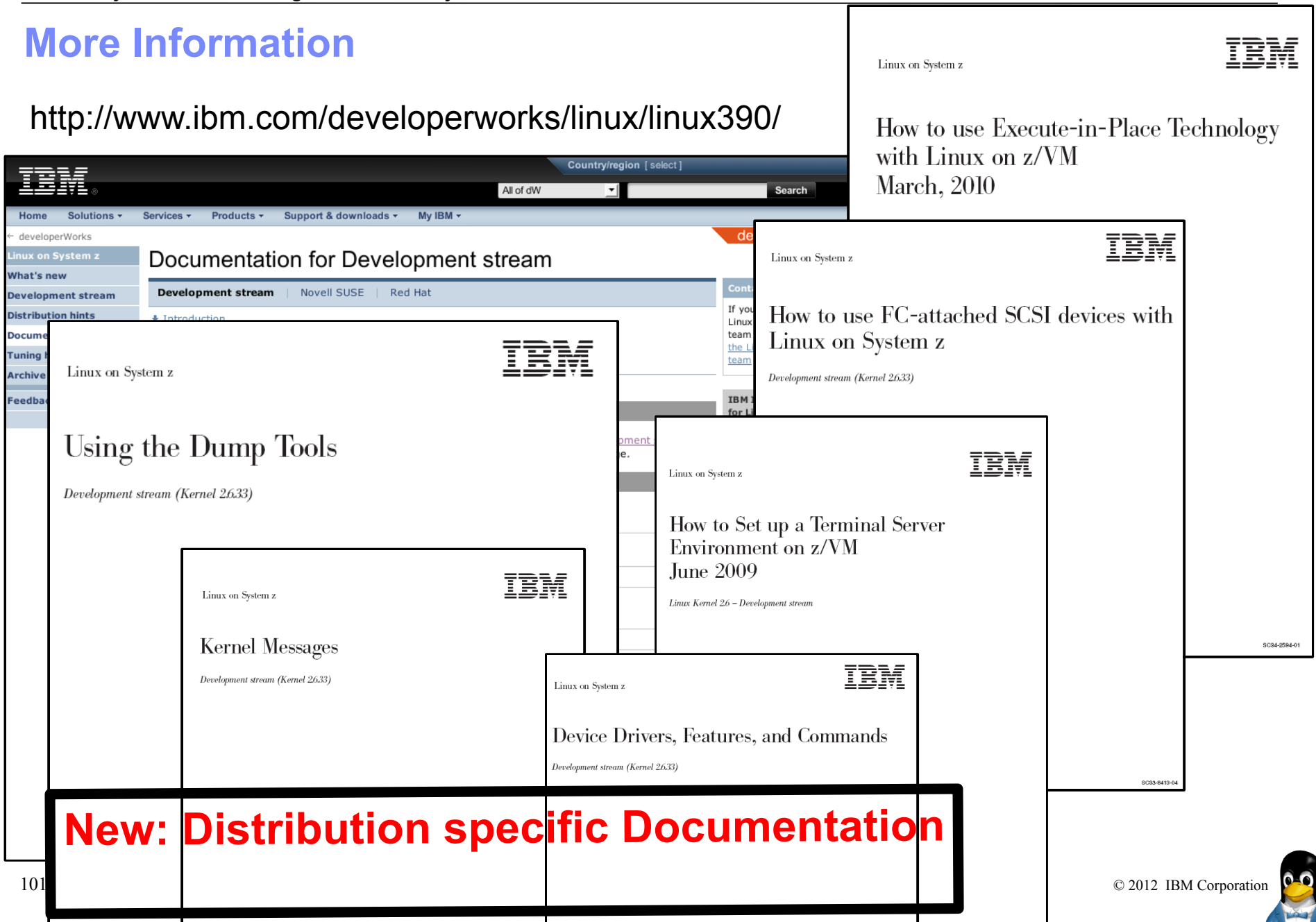
- Configuration:
  - SLES10 system, connected via OSA card and using bonding driver
- Problem Description:
  - Bonding only working with 100mbps
  - FTP also slow
- Tools used for problem determination:
  - dbginfo.sh, netperf
- Problem Origin:
  - ethtool cannot determine line speed correctly because qeth does not report it
- Solution:
  - Ignore the 100mbps message – upgrade to SLES11

bonding: bond1: Warning: failed to get speed and duplex from eth0, assumed to be 100Mb/sec and Full

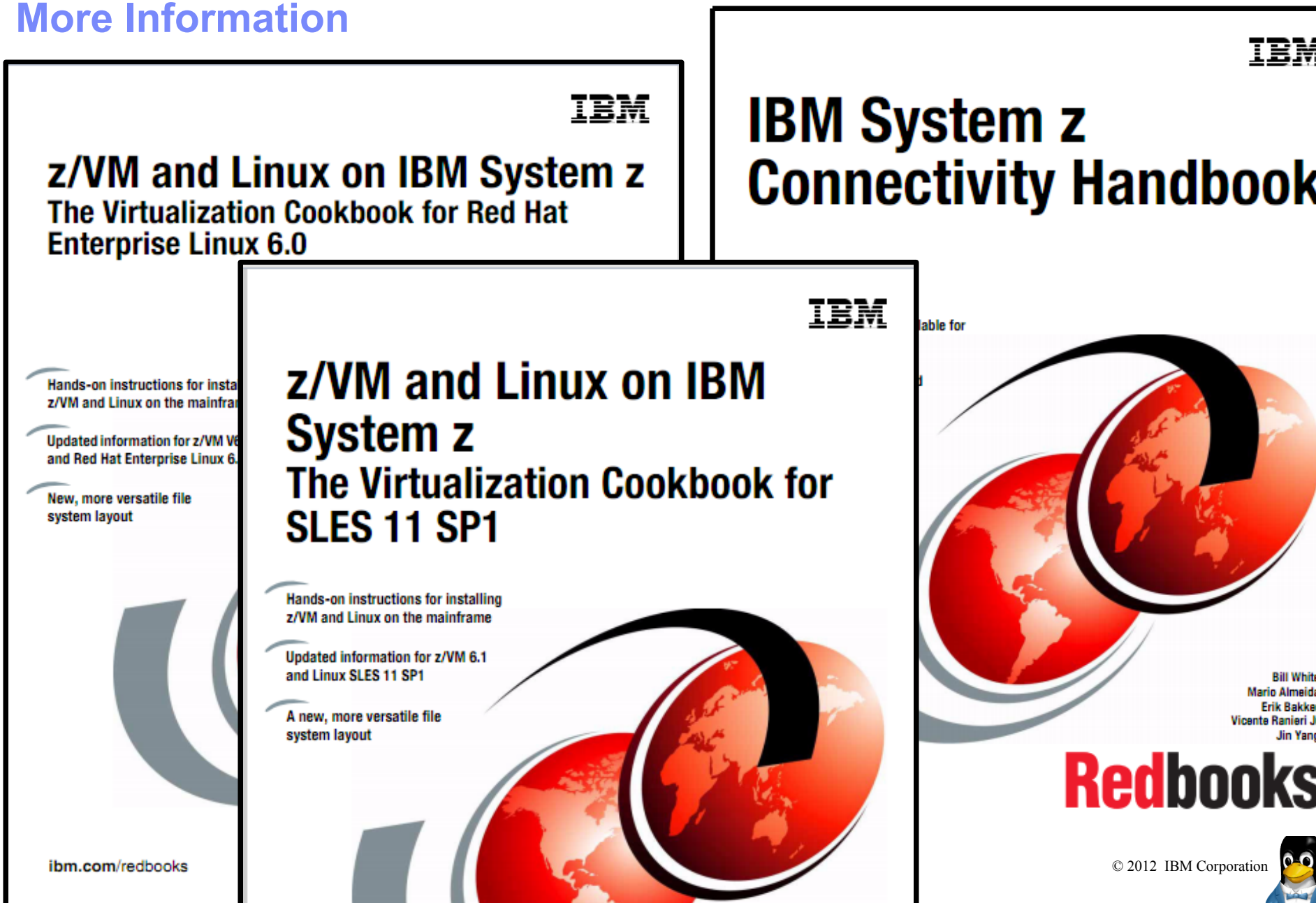
More Information

More Information

http://www.ibm.com/developerworks/linux/linux390/



More Information



References

- Linux on System z on developerWorks

http://www.ibm.com/developerworks/linux/linux390

- Linux on System z documentation

http://www.ibm.com/developerworks/linux/linux390/documentation_dev.html

- Linux on System z – Downloads

http://www.ibm.com/developerworks/linux/linux390/development_recommended.html

- Linux on System z - Tuning Hints & Tips

http://www.ibm.com/developerworks/linux/linux390/perf/index.html

- IBM System z Connectivity Handbook

http://www.redbooks.ibm.com/redpieces/abstracts/sg245444.html

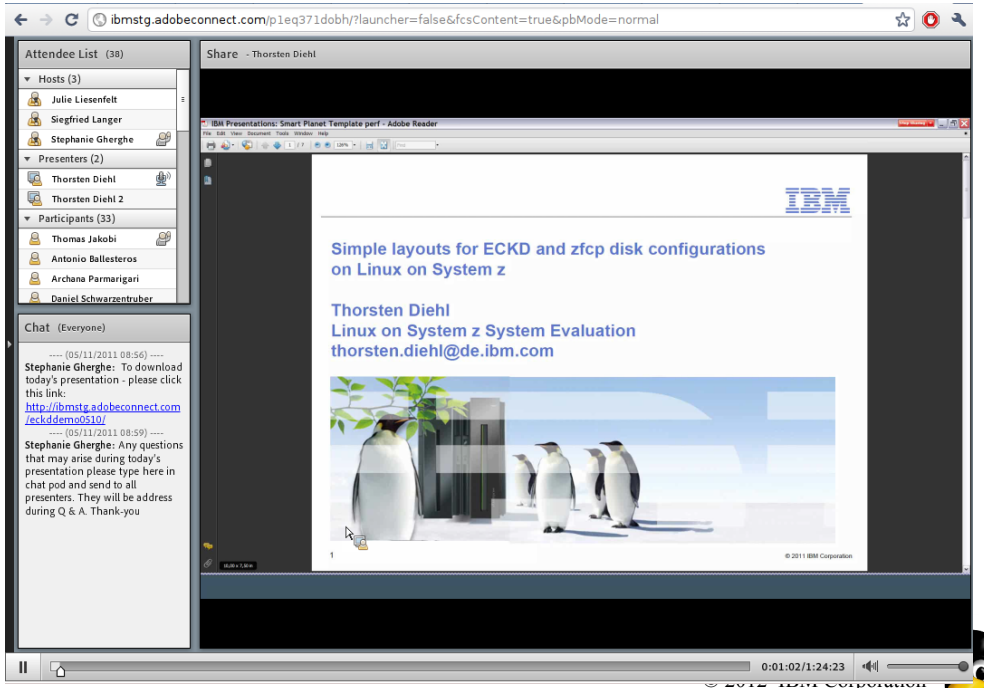
Live Virtual Classes for z/VM and Linux

http://www.vm.ibm.com/education/lvc/

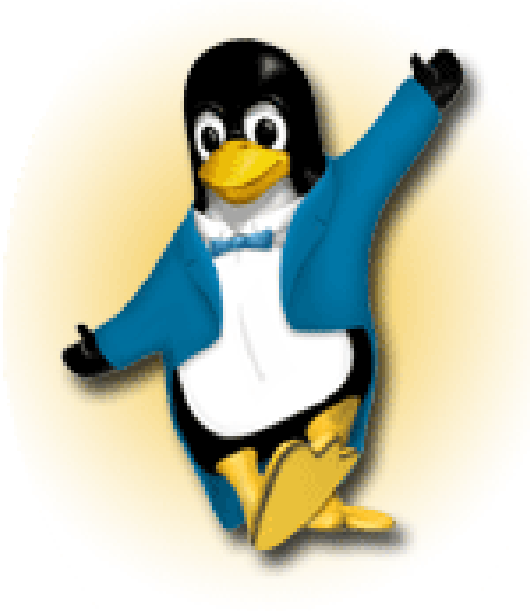
IBM offers education on a variety of z/VM, Linux on System z and z/VSE topics in the form of 'Live Virtual Classes' (LVC) available on the Internet for Customers, Business Partners and IBMers

The day of the LVC broadcast, you can see the charts and listen to the speaker 'live'. In addition, you are able (and are encouraged) to ask questions of the speaker during a Q&A session following the prepared presentation.

* The day following each LVC, we post the the charts in PDF format.
* Shortly thereafter we provide a replay where you can read the charts, hear the recording and the Q's and A's in MP3 Format
*. You are welcome to read the charts or listen to the replay without registration when you can't participate 'live' or even if you wish to hear it all again.



Questions?



Hans-Joachim Picht

Linux Technology Center

IBM Deutschland Research & Development GmbH
Schönaicher Strasse 220
71032 Böblingen, Germany

Phone +49 (0)7031-16-1810
Mobile +49 (0)175 - 1629201
hans@de.ibm.com

Your Linux on System z Requirements?

Are you missing a certain feature, functionality or tool? **We'd love to hear from you!**

We will evaluate each request and (hopefully) develop the additional functionality you need.

Send your input to hans@de.ibm.com

Thank you!
ibm.com/systems/z

Trademarks & Disclaimer

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries. For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml: AS/400, DB2, e-business logo, ESCON, eServer, FICON, IBM, IBM Logo, iSeries, MVS, OS/390, pSeries, RS/6000, S/390, System Storage, System z9, VM/ESA, VSE/ESA, WebSphere, xSeries, z/OS, zSeries, z/VM.

The following are trademarks or registered trademarks of other companies

Java and all Java-related trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries. LINUX is a registered trademark of Linux Torvalds in the United States and other countries. UNIX is a registered trademark of The Open Group in the United States and other countries. Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation. SET and Secure Electronic Transaction are trademarks owned by SET Secure Electronic Transaction LLC. Intel is a registered trademark of Intel Corporation. * All other products may be trademarks or registered trademarks of their respective companies.

NOTES: Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply. All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions. This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography. References in this document to IBM products or services do not imply that IBM intends to make them available in every country. Any proposed use of claims in this presentation outside of the United States must be reviewed by local IBM country counsel prior to such use. The information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice. Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.