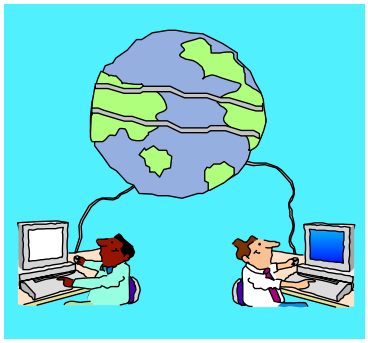
Session 8240

Securely Isolating and Segmenting Traffic across Shared OSA Ports

(CHPID Types OSD and OSX)



Thursday, March 3, 2011
4:30 PM - 5:30 PM

Room 212A
(Anaheim Convention Center)

Gwendolyn J. Dente (gdente@us.ibm.com)
z/OS Communications Server and Networking Communications Specialist
IBM Advanced Technical Skills (ATS)
Gaithersburg, MD 20879

Abstract

- Implementing security on the mainframe is a "hot topic." But people are confused about the topic of security, because it encompasses much more than encryption or providing access control lists. It can also apply to separating traffic that must be secured from other traffic that is available to anyone.
- And this is where the idea of isolating portions of the network from other parts of the network comes into play. However, if you are sharing OSA ports among multiple system images -- one of the strengths of the System z -- how can you isolate (or segment) one type of traffic from another over that shared port? A famous set of Security Mandates (Payment Card Industry mandates - "PCI") even touts the benefits of network segmentation as follows:
- "Adequate network segmentation, which isolates systems that store, process, or transmit cardholder data from those that do not, may reduce the scope of the cardholder data environment."
- This session explains shared OSA ports in terms of Virtual LANs, port isolation, routing capabilities to show how you can make a single port securely carry traffic that must be kept private while transporting other traffic that is public

Agenda

1. What is Segmentation?
2. Outbound and Inbound Routing with Shared OSA Ports
3. OSA IP Address Registration (the OAT)
4. Segmenting Traffic Across a Shared OSA Port
   - Using Virtual LANs (VLANs)
     1. Tuning for VLANs
   - Using ISOLATE
   - Using Policy Based Routing (PBR)
5. Segmenting Traffic Across a Shared OSA Port (in the Ensemble)
   - CHPID Type of OSM or OSX
   - Using Virtual LANs
     1. Ensemble and Network Virtual Management Security with VLANs

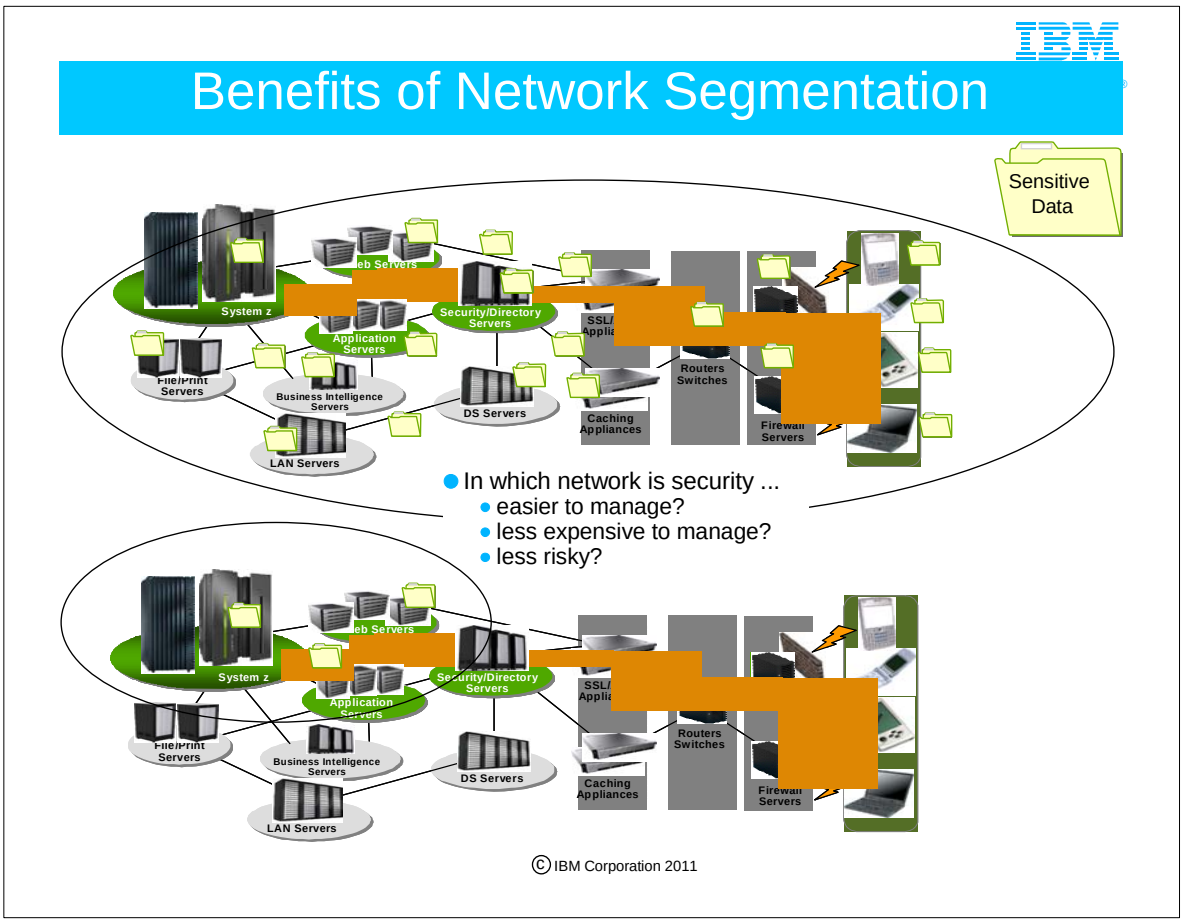
Why is Segmentation Desirable for Security Purposes?

- Segregating sensitive data, processes, traffic flow to isolated parts of the network can reduce the scope of security vulnerability.

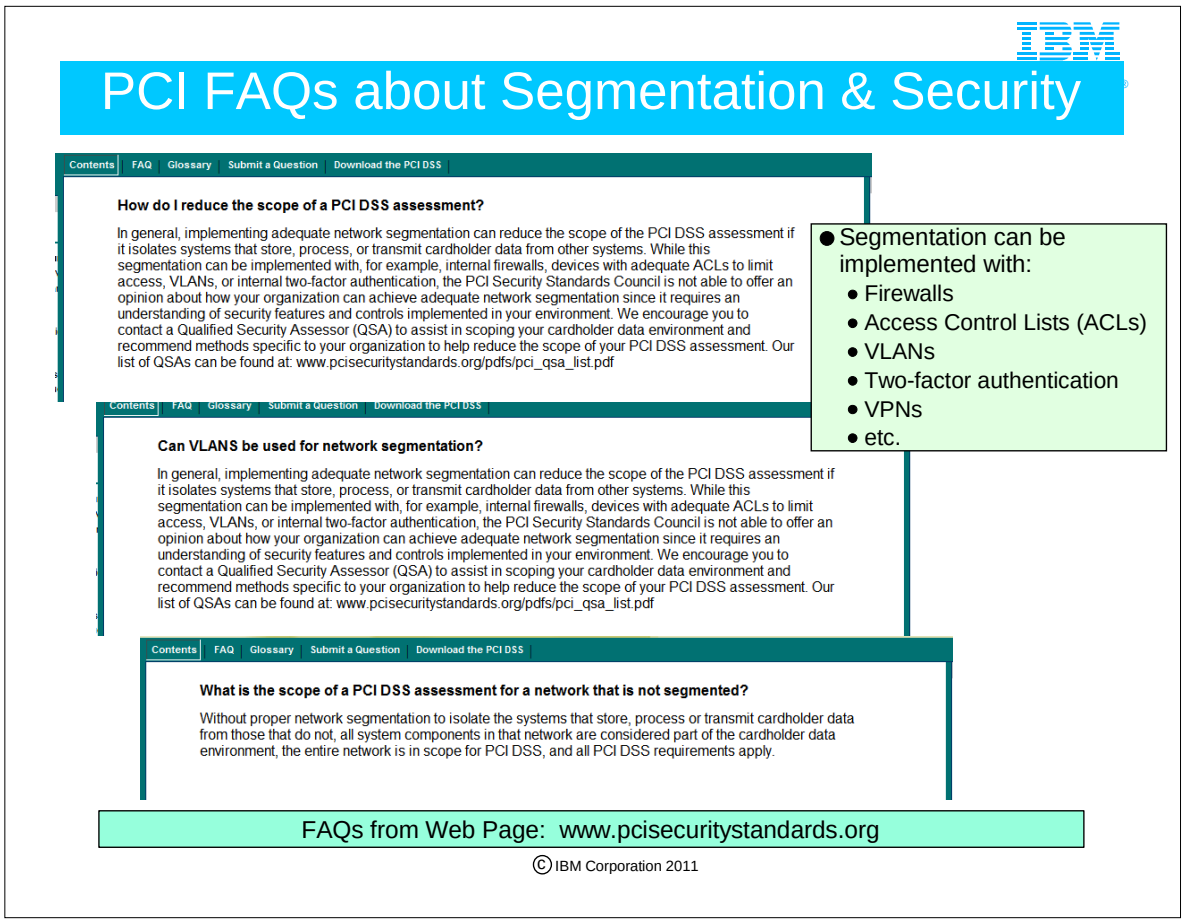
Payment Card Industry (PCI), Segmentation

- **PCI DSS requirements apply to all system components that are included in or connected to the cardholder data environment.**
- The cardholder data environment is that part of the network that possesses cardholder data or sensitive authentication data, including network components, servers and applications.
  - Network components may include but are not limited to firewalls, switches, routers, wireless access points, network appliances, and other security appliances.
  - Server types may include but are not limited to the following: web, database, authentication, mail, proxy, network time protocol (NTP), and domain name server (DNS).
  - Applications may include but not limited to all purchased and custom applications, including internal and external (Internet) applications.
- ***Adequate network segmentation, which isolates systems that store, process, or transmit cardholder data from those that do not, may reduce the scope of the cardholder data environment.***
- A Qualified Security Assessor (QSA) can assist in determining scope within an entity's cardholder data environment along with providing guidance about how to narrow the scope of a PCI DSS assessment by implementing proper network segmentation.

Reference PCI DSS V1.2:
https://www.pcisecuritystandards.org/pdfs/pci_dss_v1-2.pdf

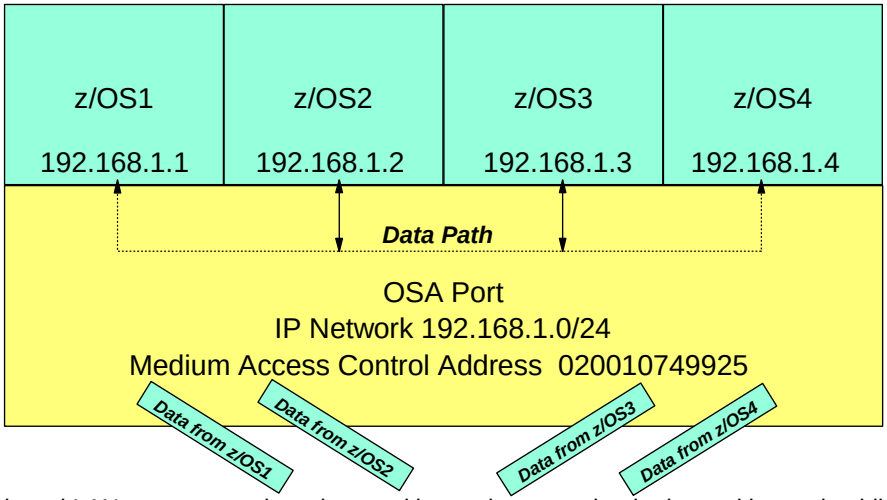


1. Network Segmentation
   1. Isolating (segmenting) the cardholder data environment from the remainder of the cardholder network in order to reduce:
      1. The scope and cost of the PCI DSS assessment
      2. The cost and difficulty of implementing and maintaining PCI DSS controls
      3. The risk to an organization (reduced by consolidating cardholder data into fewer, more controlled locations)
2. Obviously the network in the bottom half of the visual is easier to manage in all respects of security.
   1. The sensitive data has been isolated over limited network paths and onto limited systems. In other words, network segmentation has limited the scope of the vunlnerability to security breaches.



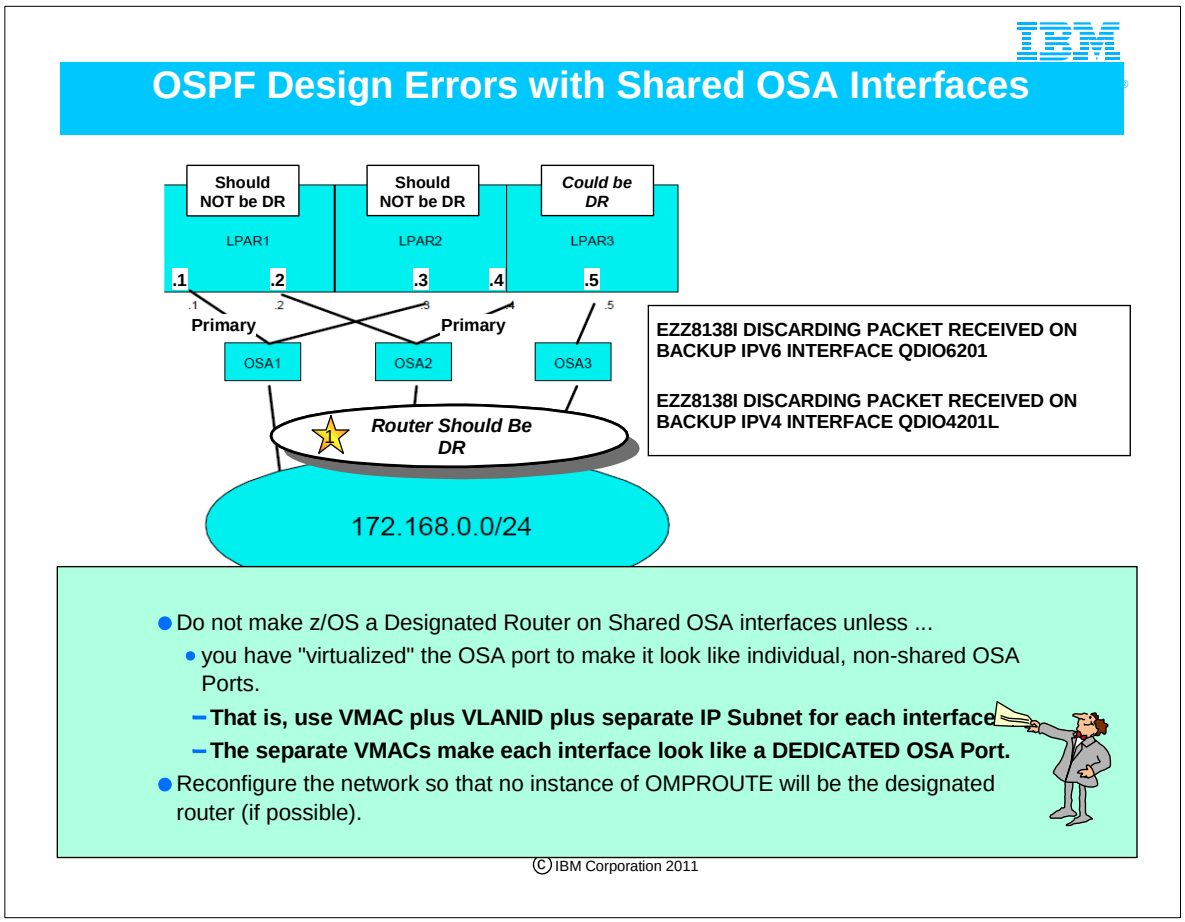
1. The Payment Carc Industry Security Standards Organization inludes comments about the desirablilty of network segmentation in the Payment Card Industry standards documents and also in the Frequently Asked Questions (FAQ) sections of their web pages.

Sharing QDIO OSA Ports on System z



- A shared LAN segment or shared networking path means that both sensitive and public data could be intermingled on the path.
  - Some auditors consider this a security exposure and require segmentation via encryption, VLANs, access control lists, and so on to protect the data.
  - Some auditors have mistakenly even expressed doubt about the security of a shared internal datapath, even when it is a virtualized path, like the
    - **HiperSockets connection or a**
    - **Shared OSA connection.**

1. A LAN adapter (i.e., the OSA Port) can be shared by multiple TCP/IP images on System z.
2. Here you see four z/OS LPARs sharing a single OSA port that resides on the IP Subnet of 192.168.1.0/24.
3. The data paths between the four z/OS Systems can be internal, passing directly through the OSA port.
4. Alternatively it is possible to force the data path among the four systems over the shared physical connection to the LAN switch and then across an external router.
5. The data path to the outside world passes over the shared port connection.
6. A shared LAN segment or shared networking path means that both sensitive and public data could be intermingled on the path.
7. Some auditors consider this sharing of LAN segment a security exposure and require segmentation via encryption, VLANs, access control lists, and so on to protect the data.
8. Some auditors have even expressed doubt about the security of a shared internal datapath, even when it is a virtualized path, like the
   1. HiperSockets connection or a
   2. Shared OSA connection.
9. We believe there is no exposure when using such a secured, internal path, but customers have often asked us to provide mechanisms to make these internal paths even more secure, by adding VLANs, or permitting tracing or sniffing on these paths.

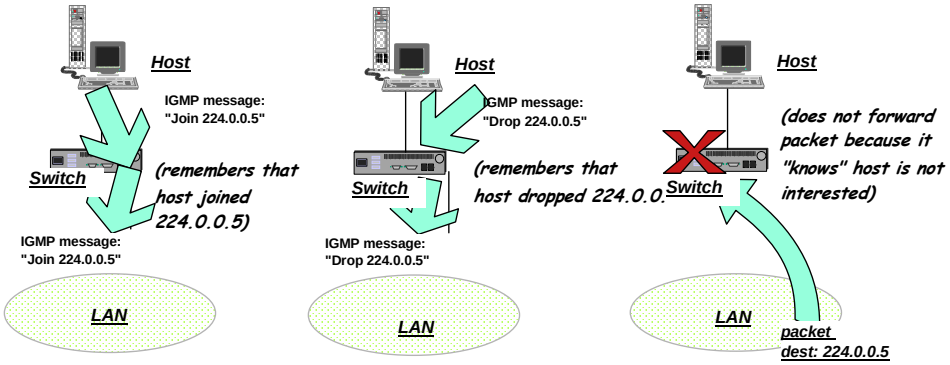


1. In the above diagram, OSA1 is shared by LPAR1 and LPAR2's 172.168.0.1 and 172.168.0.3, respectively
2. OSA2 is shared by LPAR1 and LPAR2's 172.168.0.2 and 172.168.0.4, respectively.
3. OSA3 is not shared across LPARs.
4. If, for example LPAR1 chooses 172.168.0.1 to be primary OSPF interface, and LPAR2 chooses 172.168.0.4 to be its primary OSPF interface, unicast packets can be rejected or discarded.
5. Because the OSAs are configured as QDIO, LPAR1 knows it can send a unicasted database description packet to 172.168.0.4 over the OSA card to LPAR2 without going out over the network.
6. The unicast packet is sent from LPAR1 over the card and forwarded up to the TCPIP stack on LPAR2 – however the OSA may or may not assign the correct destination address to the packet when it gets passed up to the stack. The design of LPAR-LPAR communication over QDIO was such that it doesn't matter what specific destination the packet was sent to, but what specific LPAR.
7. In the above example, the problem can be avoided by making LPAR3 the designated router. Only the designated router will exchange unicasted database description packets with other neighbors, and because OSA3 is not shared, there is not a chance to run into the problem because unicasted database description packets will not be exchanged between shared LPARs. The key to avoiding the problem is to not allow unicasted OMPROUTE traffic flow between LPARs with shared OSAs.
8. EZZ8138I message has been created to warn when unicast database description packets are discarded
    1. This EZZ8138I message is output to the console the first time a packet is discarded on an interface. Any additional discards on that interface within 5 minutes will be logged via a new debug message in the OMPROUTE trace facility. After 5 minutes the next discard will again result in a console message.
    2. Any interim discards will be logged via a new debug message in the OMPROUTE trace facility.
9. You can only be experiencing a problem with unicast packet discards if you meet all the criteria;
    1. multiple parallel OSPF interfaces in the same subnet (or link for IPv6),
    2. the designated router for this subnet is another OMPROUTE instance, and
    3. this OMPROUTE is sharing an OSA card in QDIO mode with the designated router OMPROUTE.
10. If this is your problem, there are two ways to fix it:
    1. reconfigure the network so that the OSAs are not shared between the designated router OMPROUTE's TCPIP stack and any other OMPROUTE's TCPIP stacks,
        1. If you assign separate VMACs, VLANs and separate IP Subnets to multiple interfaces into the same OSA Port, you have in essence assigned non-shared OSA Ports to the LPARs and their interfaces.
    2. reconfigure the network so that no instance of OMPROUTE will be the designated router (if possible).

Watch Out for Multicast Snooping!

Some switches offer an optimization called Multicast Snooping (aka IGMP Snooping), which is not architected in RFCs.

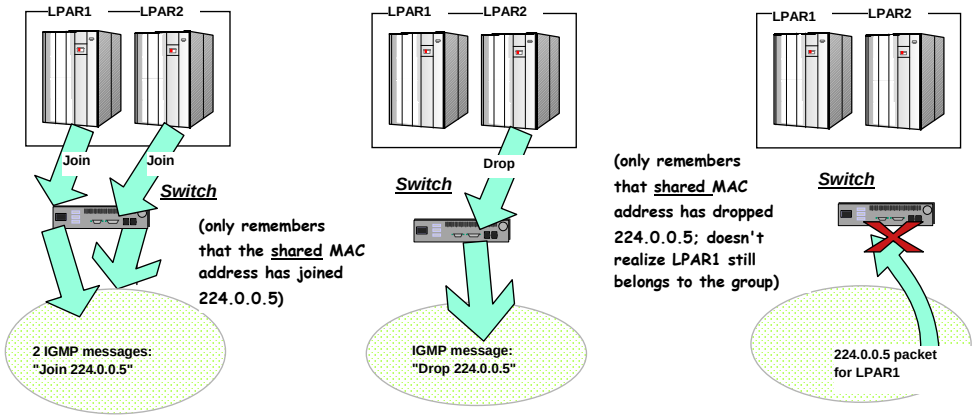
- Normally switches should forward all multicast packets to all attached hosts to let them decide if they are interested.
- But with multicast snooping, the switch listens for multicast joins and drops and does not forward multicast packets if it thinks a host is not in a multicast group.



Uh-Oh! Multicast Snooping with Shared OSAs

Switches identify hosts by MAC addresses.

- Multiple LPARs on a z/OS box which are sharing an OSA port have the same MAC address, and the switch doesn't know there are multiple "hosts"
- If the switch sees a drop of an OSPF multicast address from a MAC address, it stops forwarding of all OSPF multicast packets to that MAC address.
  - ***Therefore, if one LPAR drops the OSPF multicast group, any other LPARs on that box that are running OSPF will be crippled!***



1. This problem could be overcome if switches that implemented multicast snooping kept a count and only stopped forwarding packets for a multicast group when there have been as many drops as joins
2. However, the designers of this optimization did not do that, most likely because they did not think that multiple hosts could be using the same MAC address.
3. IGMP Snooping or Multicast Snooping can cause problems only on certain router implementations -- not on all.

Multicast (IGMP) Snooping Conclusion

Symptoms can include:

- ***Adjacencies dropping for no apparent reason***
- ***Different routers on the same LAN having different neighbor lists***
- ***More than one designated router on a LAN network***

Don't enable multicast snooping on a switch when:

- *it is attached to a z/OS box running multiple LPARs with shared OSAs, and*
- *more than one of those LPARS will be running OMPROUTE (or any application or service that uses multicast, for that matter)*

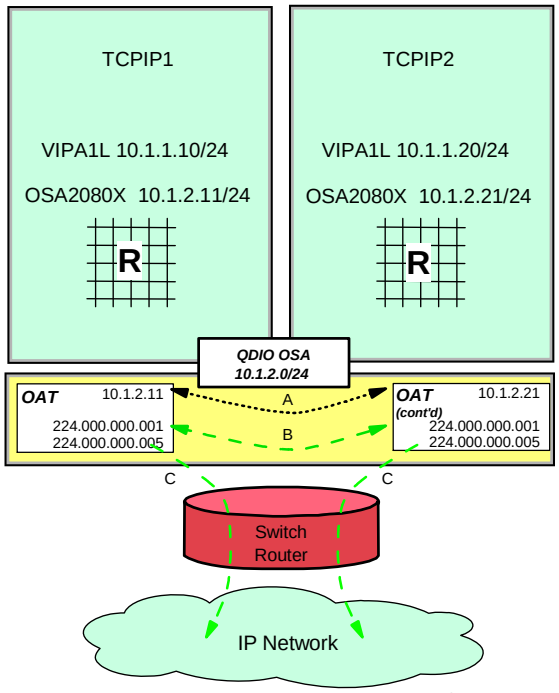
If you don't know what you're looking for, problems caused by multicast snooping can be very hard to debug

- *because the multicast packets that the switch drops don't show up at all in OMPROUTE or TCP/IP traces.*

Outbound and Inbound Routing Fundamentals on z with Shared OSA Ports

- Everything hinges on the IP Routing Table.

Outbound and Internal Routing of Traffic in a Shared OSA Port

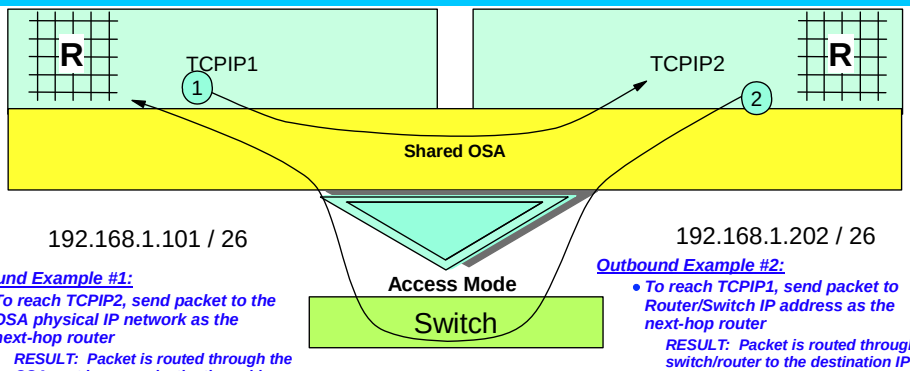


Outbound Routing

- TCP/IP stack looks in routing table.
  - If next hop is the address of the OSA port, send packet to OSA port.
    - OSA port information (registered addresses in the OSA Address Table - OAT, registered VLAN IDs) determines how to route across the OSA port to the destination.
      - A. Unicast IP Address
      - B. Multicast IP Address
  - If next hop is an IP address outside the OSA port, send packet there.
    - C. Sent to Router

1. Many customers share OSA-Express ports across logical partitions, especially if capacity is not an issue. Each stack sharing the OSA port registers certain IP addresses and multicast groups with the OSA.
2. For any traffic that is to be routed outbound from the TCP/IP stack over the OSA port, the stack looks in the routing table (indicated by "R" in the TCP/IP stack visual).
   1. If the routing table indicates that the next-hop address to the destination is the OSA port itself,then the OSA microcode examines the information frame for a VLAN header and an IP address and makes a decision on how to route the frame through the OSA port.
      1. For performance reasons, the OSA-Express bypasses the LAN and routes packets directly between the stacks when possible -- that is when the routing table points to the OSA adapter port as the next hop to the destination and when the information registered in the OSA port indicates that the information can be sent across the port..
   2. If the routing table indicates that the next hop address to the destination is a router in the external network, the packet is routed to that router using the MAC address of the external node. Note: The MAC address is learned through the Address Resolution Protocol (ARP) of the TCP/IP architectural flows.
3. For unicast packets, OSA internally routes the packet when the next-hop IP address is registered on the same LAN or VLAN by another stack sharing the OSA port.
   1. A: You see how TCPIP1 routes a packet to 10.1.2.21 in TCPIP2 over the OSA port without exiting out onto the LAN because the next hop to reach the destination is registered in the OSA Address Table (OAT); the TCPIP1 routing table indicates that the destination can be reached by hopping through the direct connection to the 10.1.2.0/24 network.
   2. B For multicast (e.g., OSPF protocol packets), OSA internally routes the packet to all sharing stacks on the same LAN or VLAN which registered the multicast group. Note how TCPIP1 and TCPIP2 have each registered multicast addresses for OSPF (224.000.000.00n) in the OSA port.
   3. C OSA also sends the multicast/broadcast packet to the LAN. For broadcast (not depicted), OSA internally routes the packet to all sharing stacks on the same LAN or VLAN.

OSAs: When is traffic routed through the OSA?



Outbound Example #1:

- ***To reach TCPIP2, send packet to the OSA physical IP network as the next-hop router***
  - ***RESULT: Packet is routed through the OSA port because destination address is in the OSA Address Table (OAT)..***

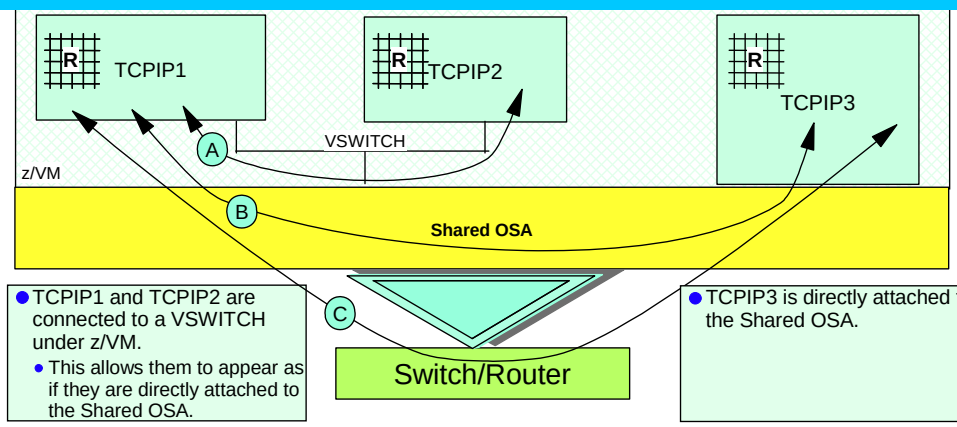
Outbound Example #2:

- ***To reach TCPIP1, send packet to Router/Switch IP address as the next-hop router***
  - ***RESULT: Packet is routed through the switch/router to the destination IP address..***

It all depends on the layer three routing table!

- Customer coded only a default route at TCPIP2 and did not verify if other, direct-connected routes were available. All the traffic from TCPIP2 to TCPIP1 was routed through the switch/router.
  - Moral #1: Always display routing table at the operating system to determine if the routes you need have been built or not.
    - Notes:
      - DYNAMICXCF in z/OS automatically builds routing table entries for the XCF, HiperSockets, and IUTSAMEH address.
      - Linux automatically builds direct-connected routes for devices defined to it.
      - VM automatically builds direct-connected routes for devices defined to it.
  - Moral #2: You may need to provide static routes or dynamic routes to create the routing paths you want.

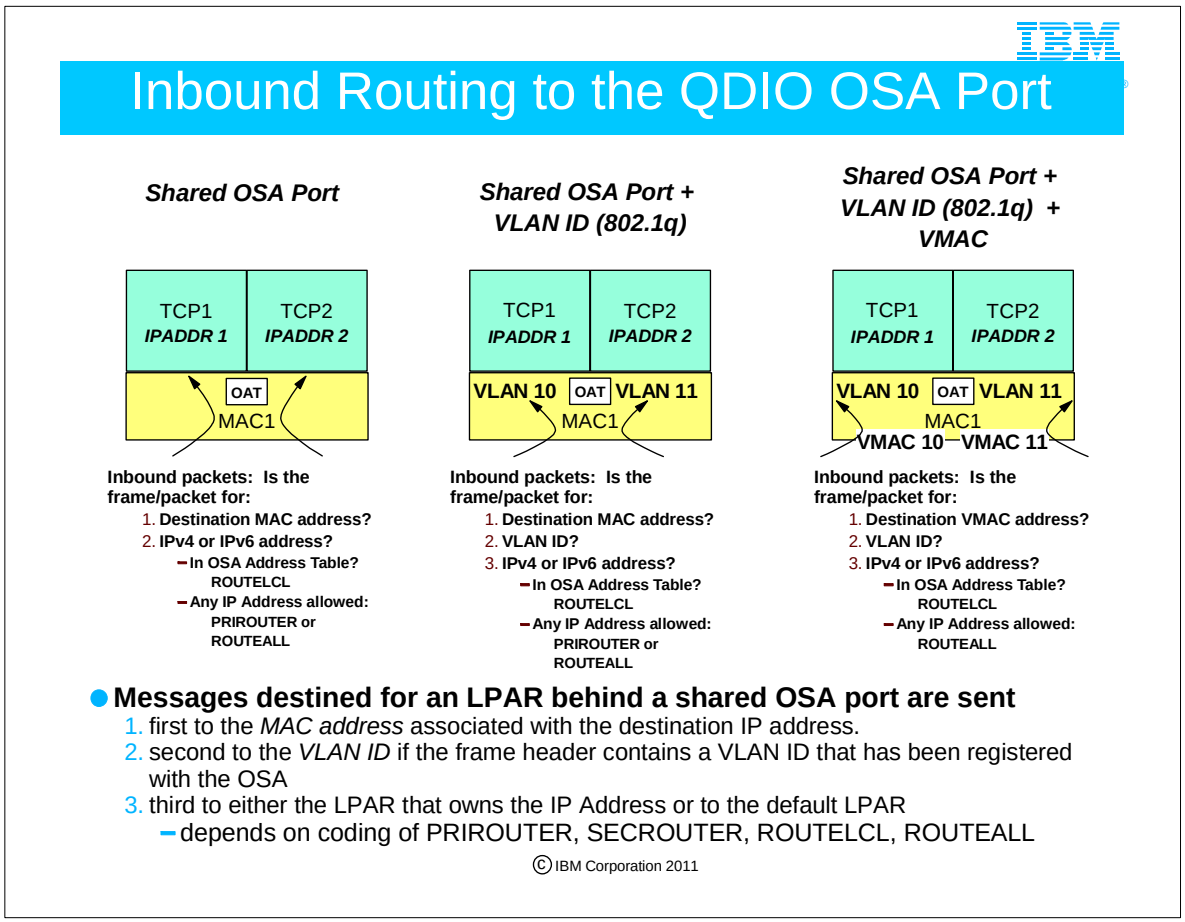
Routing Paths when a VSwitch is Employed



- TCPIP1 and TCPIP2 are connected to a VSWITCH under z/VM.
  - This allows them to appear as if they are directly attached to the Shared OSA.
- TCPIP3 is directly attached to the Shared OSA.

A. If TCPIP1 communicates with TCPIP2, it is likely to use the path (A) over the VSWITCH path in z/VM memory.
B. If TCPIP1 communicates with TCPIP3, it is likely to use the path (B) over the Shared OSA.
C. However, the routing table in TCPIP1 can cause the path to TCPIP3 to lead through the external Switch/Router (C).

The path taken depends on the structure of the IP stack's layer three routing table!



1. The routers on the LAN always send any packets destined for a particular TCP/IP stack to the VMAC defined for that stack. The OSA-Express feature knows by VMAC address exactly which stack should receive a given packet. Even if the IP address is not registered with the OSA-Express feature, if the packet is destined for that VMAC, the router has determined which stack should be the intermediate router, and the OSA can forward the packet directly to that stack. If the stack is not an intermediate router, the capability is provided for a stack to indicate to the OSA that it wants to receive packets to registered IP addresses only.
2. Effect of PRIROUTER and SECROUTER with and without VLAN
   1. The VLANID parameter of the LINK and INTERFACE statements interacts with the PRIRouter and SECRouter parameters on the DEVICE and INTERFACE statements.
   2. If you configure both a VLANID and either PRIRouther or SECRouter, then this TCP/IP instance will act as a router for this VLAN only. Frames that are received at this device for an unknown IP address will only be routed to this TCP/IP instance if they are VLAN tagged with this VLAN ID.
   3. If you do not configure a VLAN ID, but do configure PRIRouter or SECRouter, then this TCP/IP instance will act as a "Global default Pri/SecRouter" for all inbound unicast frames with an unknown destination IP address
3. Adding a Virtual MAC (VMAC) to a LAN Connectin:
   1. Enables first hop routers and load balancers to use dispatch mode (MAC-level) forwarding
   2. Avoids use of GRE
   3. Enables use of dispatch mode by devices that do not support GRE (Cisco CSM and CSS)
   4. Enables use of dispatch mode for IPv6 for which GRE isn't defined
   5. Removes the need for using NAT instead of dispatch mode forwarding
   6. NAT requires strict control of outbound path to handle NAT on outbound flows

4. Makes System z LPARs look more like "normal" TCP/IP nodes on a LAN
   1. Simplifies network infrastructure
   2. Avoids the whole PRIROUTER/SECROUTER setup issue when sharing a port between multiple LPARs
   3. Layer-2 visibility into final source/destination of LAN traffic
   4. Filter z/OS LAN frames in "sniffer"-type devices by MAC addresses instead of layer-3 information
5. On an OSA-E2, VMAC was introduced:
   1. System z9 with OSA-Express or OSA-Express2 and z/OS V1R8 with PTFs or z/OS V1R9

OSA Address Table (OAT) Registration Details for z/OS

Single Connection per Stack per Subnet (2)

Connection Definition Type	Subnet coded?	Basic definition (no VMACs)	VMAC ROUTEALL configured? *For routing: IP addresses are ignored and routing is based solely on VMAC*		VMAC ROUTELCL configured? *For routing: IP addresses used for routing once packet has reached the VMAC*
DEVICE and LINK (2)	N/A	Most active IP addresses in HOME List -- OSA (3) -- CTC -- XCF -- HiperSockets, etc. -- All VIPAs	**V1R11 & Prior:** Most active IP addrs. in HOME List: (1) -- OSA (3) -- CTC -- XCF -- HiperSockets, etc. -- All VIPAs	**V1R12 & Later (6):** --OSA IP address (3) --All VIPAs (4)	Most active IP addresses in HOME List -- OSA (3) -- CTC -- XCF -- HiperSockets, etc. -- All VIPAS
INTERFACE (2)	No	Most active IP addresses in HOME List -- OSA (3) -- CTC -- XCF -- HiperSockets, etc. -- All VIPAs	-- OSA IP address (3) -- All VIPAs (4)		Most active IP addresses in HOME List -- OSA (3) -- CTC -- XCF -- HiperSockets, etc. -- All VIPAs
INTERFACE (2)	Yes	Most active IP addresses in HOME List -- OSA (3) -- CTC -- XCF -- HiperSockets, etc. -- VIPAs in same subnet as OSA (4) -- All other VIPAs (5)	-- OSA IP address (3) -- VIPAs in same subnet as OSA (4)		Most active IP addresses in HOME List -- OSA (3) -- CTC -- XCF -- HiperSockets, etc. -- VIPAs in same subnet as OSA (4) -- All other VIPAs (5)

General NOTES: z/OS registers IP addresses for two purposes: for inbound routing and for ARP offload. The OSA/SF "GET OAT" function displays only addresses that have been registered for ARP offload purposes.The coding of VLANID does not influence ARP registration; it does, however, influence routing; if VLAN is coded on the z/OS or z/VM stack, only correctly tagged packets are considered for routing, after which the IP address will be considered for VMAC ROUTELCL and ignored for VMAC ROUTEALL.

Note (1): Since routing in this case is based solely on VMAC, the registration of all IP addresses for routing purposes is superfluous. If ROUTEALL is coded, the OSA will route any received packets to this stack if the destination MAC address matches the coded VMAC, regardless of which addresses are registered.

Note (2): If a stack has multiple connections to the same IP subnet or VLAN, the VIPAs are registered for ARP offload purposes with only one of the multiple OSA connections.

Note (3): If ARP takeover is in effect, the OSA address that we have taken over is also registered for ARP offload purposes

Note (4): Registered for ARP offload purposes (sends gratuitous ARPs and responds to ARP requests)

Note (5): Registered for inbound routing

Note (6): The change in V1R12 avoids registrations for inbound routing which serve no purpose

1. Although this chart is built to indicate that we are describing the OAT behavior when only a single connection per stack exists, the chart applies also to the case of multiple connections with one exception.
    1. if we need to register for inbound routing (i.e. we are not VMAC ROUTEALL), then we register the VIPAs ARP-yes on one intf and ARP-no on the other.....if we are VMAC ROUTEALL, then we don't register anything ARP-no
    2. With VMAC ROUTEALL, one OSA port would see ARP-yes for the Registered VIPA in the same subnet as the OSA port; the other OSA port(s) would not even have received a registration for the VIPA. Without VMAC ROUTEALL, one OSA port would see ARP-yes for the Registered VIPA in the same subnet as the OSA port; the other OSA port(s) would have received a registration for the VIPA but it would show up as ARP-no.
2. Generally speaking for DEVICE/LINK definitions:,
    1. If no VMAC is coded, the OSA OAT registers all addresses with the exception of hidden dynamic VIPAs and LOOPBACK addresses.
    2. If VMAC ROUTEALL is coded for a DEVICE/LINK definition, then everything but hidden dynamic VIPAs and LOOPBACK addresses is registered, but routing ignores the registered addresses and uses only the VMAC for sending a packet to the correct stack.
    3. If VMAC ROUTELCL is coded for a DEVICE/LINK definition, then everything but hidden dynamic VIPAs and LOOPBACK addresses is registered. The VMAC must be correct AND the IP address must be in the OAT for the packet to be forwarded to the correct stack.
3. Generally speaking for INTERFACE definitions:
    1. If no VMAC is coded and no IP Subnet is coded, OSA OAT registers everything just as with the DEVICE/LINK statements.
    2. If no VMAC is coded, but you have coded an IP Subnet on the INTERFACE statement, then the only VIPAs registered are the ones in the same subnet as the native HOME address of the OSA port.
    3. If VMAC ROUTEALL and no IP Subnet are coded for an INTERFACE definition, we only register the OSA IP address and all the VIPAs.
    4. If VMAC ROUTEALL and an IP Subnet are coded for an INTERFACE statement, then the only VIPAs registered are the ones in the same subnet as the native HOME address of the OSA port.
    5. Generally speaking for INTERFACE, if multiple connections to the same subnet from a single stack exist, then the OSA OAT registration registers the VIPAs with only one of the multiple interfaces.

In reality, the z/OS registers IPv4 addresses in these fashions for two distinct purposes:

1. Inbound routing
    1. "For Inbound Routing" means that the OSA port will not send gratuitous ARPs and the OSA port will also not respond to ARP requests. But the OSA port will pass any packets with a matching destination IP up to the stack depending on the conditions named in the table above.
2. ARP offload
    1. "For ARP offload" means that the OSA port will send gratuitous ARPs and the OSA port will also respond to ARP requests. Depending on the coding for the OSA port connection, the OSA may also pass any packets with a matching destination IP up to the stack if necessary or will rely only on VMAC ROUTEALL to pass the packet to the stack..

What we have described above in terms of registration can be described in more detailed terms, splitting the registration types into two different groups: the addresses registered for the purposes of Inbound Routing and those registered for the purposes of ARP offload. Here are the details of this view of IPv4 address registration:

1. **Inbound routing**
    1. For INTERFACE statement with VMAC ROUTEALL, we do not register any IP addresses for the purpose of inbound routing. That is, we only register IP addrs for the purpose of supporting ARP offload.
    2. For INTERFACE without VMAC ROUTEALL or for DEV/LINK, we register the entire home list for the purpose of inbound routing. (Note: for DEV/LINK with VMAC ROUTEALL, this registration is extraneous, but it doesn't hurt.)
2. **ARP offload**
    1. We always register the home IP address for the purpose of ARP offload.
    2. If you have multiple OSAs on the same (V)LAN or Physical Network (PNET), and ARP takeover is in effect, then we register the IP address of the interface for which we are taking over connection responsibility..
    3. We also register VIPAs for ARP offload purposes as follows:
        1. For the INTERFACE statement with subnet mask configured on the statement, we register only the VIPAs which are in the same subnet as the OSA.
        2. For the INTERFACE statement without a subnet mask coded on it, or for DEV/LINK, we register all the active VIPAs in the Home list
        3. (Note: For both of the above bullets, if there are multiple OSAs on same (V)LAN or Physical Network (PNET), we register these VIPAs on only one of the OSAs.

Helpful Displays to Understand Behavior of Your OSA Ports

Displaying the OSA Information (OSD)

```
D TCPIP,,OSAINFO,INTFNAME=OSX2300
EZZ0053I COMMAND DISPLAY TCPIP,,OSAINFO COMPLETED SUCCESSFULLY
EZD0031I TCP/IP CS V1R12  TCPIP Name: TCPIP     15:06:03 203
Display OSAINFO results for IntfName: OSX2300
PortName: IUTXP018     PortNum: 00     Datapath: 2302   RealAddr: 0002
PCHID: 0590         CHPID: 18      CHPID Type: OSX     OSA code level: 0D0A
Gen: OSA-E3         Active speed/mode: 10 gigabit full duplex
Media: Multimode Fiber         Jumbo frames: Yes   Isolate: No
PhysicalMACAddr: 001A643B2135   LocallyCfgMACAddr: 000000000000
Queues defined Out: 4  In: 1    Ancillary queues in use: 0
Connection Mode: Layer 3    IPv4: Yes   IPv6: No
SAPSup: 000FF603               SAPEna: 0008A603
IPv4 attributes:
  VLAN ID:   99        VMAC Active: Yes
  VMAC Addr: 02BECB000002   VMAC Origin: Cfg    VMAC Router: All
  AsstParmsEna: 00200C57   OutCkSumEna: 0000001A  InCkSumEna: 0000001A
Registered Addresses:
  IPv4 Unicast Addresses:
    ARP: Yes  Addr: 172.30.99.1
    Total number of IPv4 addresses:      1
  IPv4 Multicast Addresses:
    MAC: 01005E000001  Addr: 224.0.0.1
    Total number of IPv4 addresses:      1
23 of 23 lines displayed
End of report
```

Displaying the OSA Information

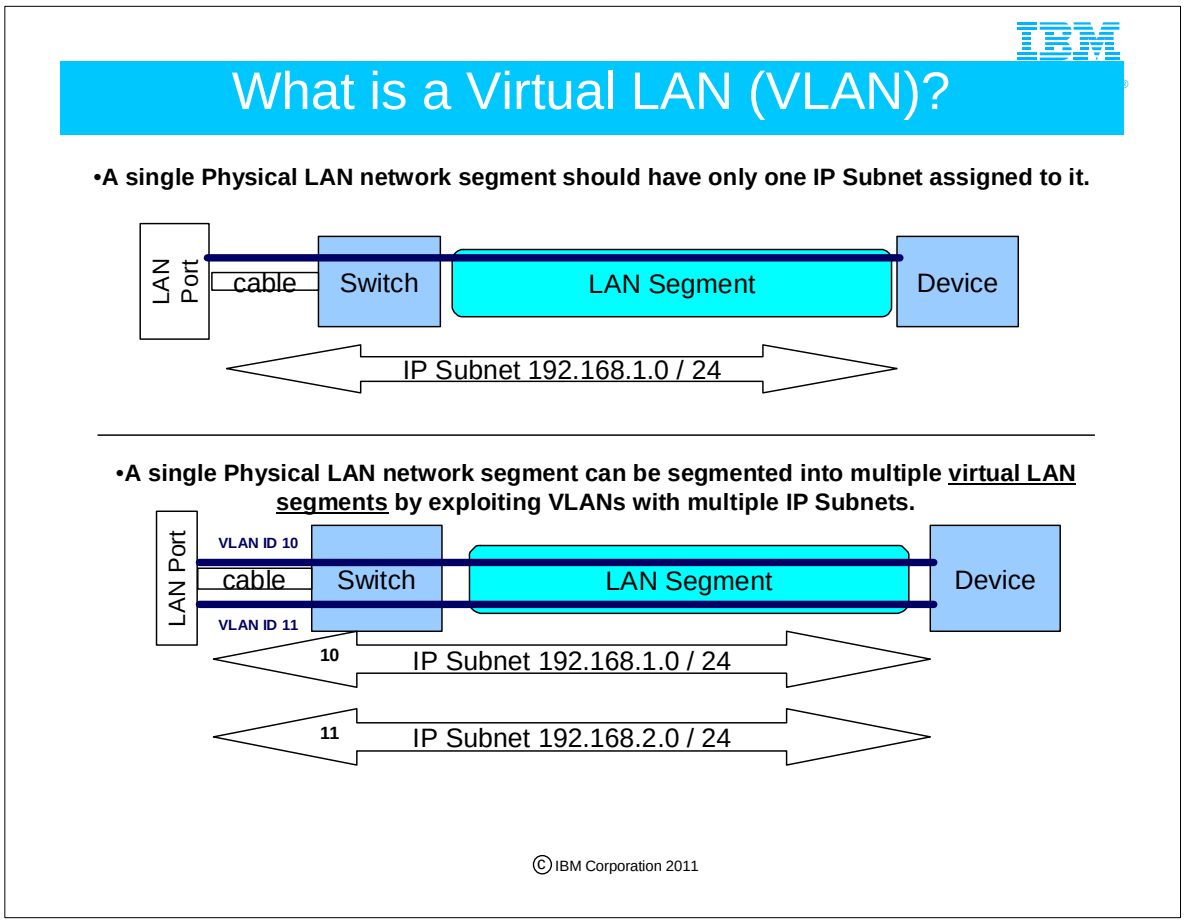
```
DISPLAY TCPIP,,OSAINFO,INTFNAME=LNK29D,MAX=500
EZD0031I TCP/IP CS V1R12 TCPIP Name: TCPCS 15:14:15
Display OSAINFO results for IntfName: LNK29D
PortName: DEV29D PortNum: 01 Datapath: 3902 RealAddr: 0002
PCHID: 0451 CHPID: 29 CHPID Type: OSD OSA code level: 6760
Gen: OSA-E3 Active speed/mode: 1000 mb/sec full duplex
Media: Singlemode Fiber Jumbo frames: Yes Isolate: No
PhysicalMACAddr: 643B88F30000 LocallyCfgMACAddr: 000000000000
Queues defined Out: 4 In: 3 Ancillary queues in use: 2
Connection Mode: Layer 3 IPv4: Yes IPv6: No
SAPSup: 00010293 SAPEna: 00010293
IPv4 attributes:
VLAN ID: N/A VMAC Active: No
Defined Router: Non Active Router: No
AsstParmsEna: 00215C66 OutCkSumEna: 00000000 InCkSumEna: 00000000
Registered Addresses:
IPv4 Unicast Addresses:
ARP: Yes Addr: 10.10.10.10
Total number of IPv4 addresses: 1
IPv4 Multicast Addresses:
MAC: 01005E000001 Addr: 224.0.0.1
Total number of IPv4 addresses: 1
Ancillary Input Queue Routing Variables:
Queue Type: BULKDATA Queue ID: 2 Protocol: TCP
Src: 11.1.1.11..100
Dst: 12.12.12.12..100
Src: 13.3.3.13..101
Dst: 14.14.14.14..101
Total number of IPv4 connections: 2

Queue Type: SYSDIST Queue ID: 3 Protocol: TCP
Addr: 10.10.10.10
Total number of IPv4 addresses: 1
33 OF 33 Lines Displayed
End of report
```

Displaying the OSA Device

```
D TCPIP,,N,DEV,INTFNAME=OSX2300
EZD0101I NETSTAT CS V1R12 TCPIP 216
INTFNAME: OSX2300           INTFTYPE: IPAQENET   INTFSTATUS: READY
    PORTNAME: IUTXP018        DATAPATH: 2302       DATAPATHSTATUS: READY
    CHPIDTYPE: OSX            CHPID: 18
    SPEED: 0000010000
    IPBROADCASTCAPABILITY: NO
    VMACADDR: 02BECB000002    VMACORIGIN: OSA     VMACROUTER: ALL
    ARPOFFLOAD: YES                  ARPOFFLOADINFO: YES
    CFGMTU: 8992                     ACTMTU: 8992
    IPADDR: 172.30.99.1/24
    VLANID: 99                      VLANPRIORITY: DISABLED
    DYNVLANREGCFG: NO                DYNVLANREGCAP: YES
    READSTORAGE: GLOBAL (4096K)
    INBPERF: DYNAMIC
      WORKLOADQUEUEING: NO
    CHECKSUMOFFLOAD: YES
    SECCLASS: 255                    MONSYSPLEX: NO
    ISOLATE: NO                      OPTLATENCYMODE: NO
 MULTICAST SPECIFIC:
    .....................(lines omitted)
 INTERFACE STATISTICS:
  .....................(lines omitted)
IPV4 LAN GROUP SUMMARY
LANGROUP: 00008
  NAME              STATUS      ARPOWNER      VIPAOWNER
  ----              ------      --------      ---------
  OSX2300           ACTIVE      OSX2300       YES
1 OF 1 RECORDS DISPLAYED
END OF THE REPORT
```

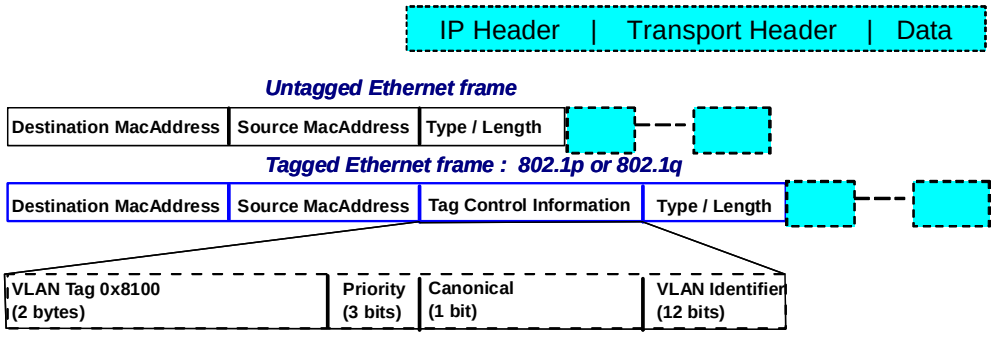
Understanding VLANs when Sharing OSA Ports on z

- Use Virtual LANs to segment a Physical LAN port into multiple Logical LAN Ports.



1. A local area network (LAN) is a broadcast domain. Nodes on a LAN can communicate with each other without a router, and nodes on different LANs need a router to communicate. A virtual LAN (VLAN) is a configured logical grouping of nodes using switches. Nodes on a VLAN can communicate with each other as if they were on the same LAN, and nodes on different VLANs need a router to communicate. (Layer 3 routers can add, remove, or validate VLAN tags.) The IBM Open Systems Adapter provides support for IEEE standards 802.1p/q, which describes priority tagging and VLAN identifier tagging. Deploying VLAN IDs allows a physical LAN to be partitioned or subdivided into discrete virtual LANs. This support is provided by the z/OS TCP/IP stack and the OSA-Express feature in QDIO mode. When you use VLAN IDs, the z/OS TCP/IP stack can have multiple connections to the same OSA-Express feature. One connection is allowed for each unique combination of VLAN ID and IP version (IPv4 or IPv6).
2. Note in the top half of the visual how one network takes advantage of the physical connectivity. In the bottom half of the visual, we have split the physical LAN into two VLANs: one with VLAN ID of 10 and another with VLAN ID of 11.
3. Z/OS and z/VM are implemented with a VLAN technology called “Global VLAN.” With Global VLAN, z/OS and z/VM can define a VLAN ID which is then registered in the OSA port. The OSA port then performs the VLAN tagging. The implementation of Global VLAN causes the stacks to be technically unaware of the VLAN, or “vlan-unaware.” However, many people find this subtle distinction confusing and refer to z/OS and zVM as “vlan-aware” stacks since they can define a VLAN ID for a LAN connection. For Linux on z (native), the TCP/IP stack itself performs the VLAN tagging, and, thus, Linux on z when running native is not using the Global VLAN ID but rather the standard 802.1q implementation of VLAN. Linux on z is thus technically a “vlan-aware” stack. You may read about Global VLAN IDs at:
4. http://publib.boulder.ibm.com/infocenter/zvm/v5r3/index.jsp?topic=/com.ibm.zvm.v53.hcpa6/hcsc9b2131.htm
5. “A GLOBAL VLAN ID is OSA's VLAN support to provide access to a virtual LAN segment for a VLAN unaware host so the host can receive and send its network traffic. This host does not tag its outbound frames nor receive tagged inbound frames. The GLOBAL VLAN ID participates on the VLAN transparently with OSA handling all the tagging work (VLAN-unaware). A host device driver can register a Global VLAN ID with the OSA-Express adapter. Typically each host defines only one Global VLAN ID per connection. Some device drivers allow configuration of one VLAN ID for IPv4 and a second VLAN ID for IPv6. The OSA-Express will use the Global VLAN ID to tag frames and send out Gratuitous ARP requests (ARP requests to check for duplicate IP addresses) on behalf of the host. The NIC simulation in z/VM also provides this support, which is separate from the virtual switch support. The Global VLAN ID processing for the virtual NIC is performed prior to any virtual switch port ingress processing and after virtual switch port egress processing.
6. One example of this is in z/VM. You can specify the VLAN keyword on a LINK configuration statement for a QDIOETHERNET link to register a Global VLAN ID.
7. To reduce complexity and host TCP/IP configuration changes when configuring a virtual switch host connection, it is recommended that you do not configure a global VLAN ID for a host that will be connected to a trunk port. Instead, connect the host to an access port and authorize it for the desired VLAN ID. This assigns a port VLAN ID (pvid) for the access port and all VLAN operations occur within the virtual switch.”

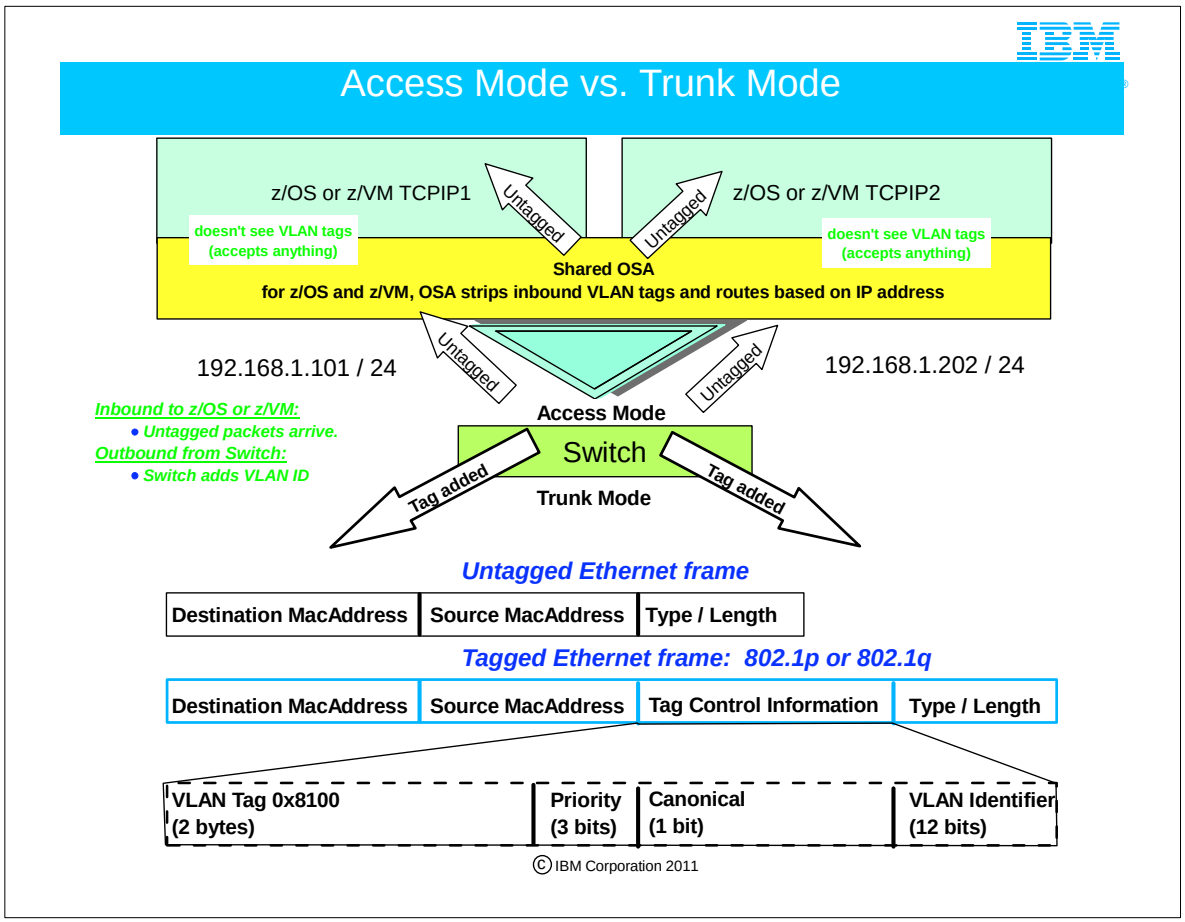
VLAN Concepts: The VLAN ID



TCP/IP Stack Architectural Layers

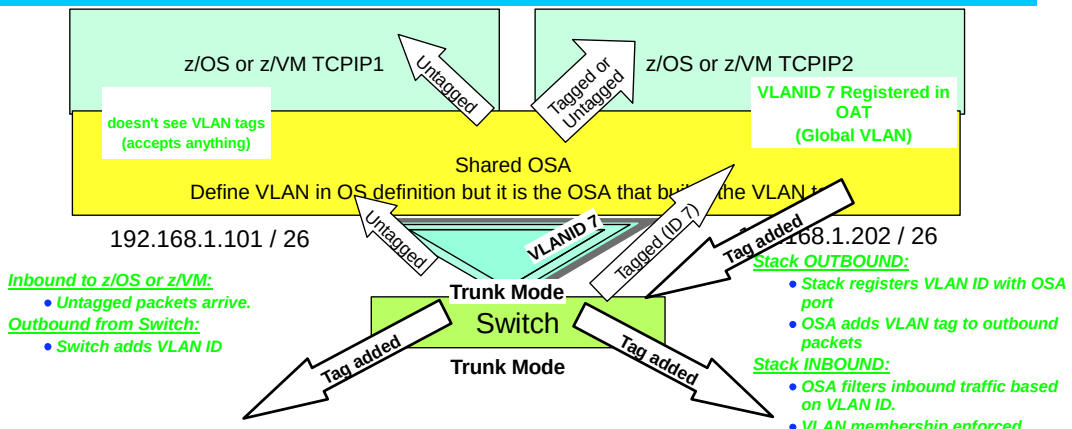
5. Application Layer	
4. Transport Layer	TCP, UDP Port Number
3. IP Routing Layer	IP Address
2. Data Link Control Layer	***Physical Address (MAC Address)*** ***VLAN ID***
[1. Physical Layer]	

1. This chart is depicting the layout of a frame that carries the data to a node on a LAN.
2. It also depicts the TCP/IP Architectural Layers and shows where VLANs and VMACs fit into the architecture.
   1. Technically Layer 1, the Physical Layer, is assumed in the architectural descriptions of the TCP/IP Architecture. The description of TCP/IP in the RFCs and the literature usually talks about “4” layers, beginning with Layer 2.
3. The data itself is packaged in an IP Packet. The IP Packet Header contains the target and the source IP Addresses. The Transport Header contains the port number of the applications that are to receive and send the data. However, to send the data across a network, the IP packet must be prefaced with a Frame Header which contains the Physical address – called the Medium Access Control Address (MAC Address), which is managed at Layer 2 of the TCP/IP stack. The LAN port that is represented by a MAC Address can further segment the data that is arriving at it by examining the VLAN Tag, which contains a VLAN ID. When using a VLAN ID to route data, this is called “Layer 2 Routing.”
4. AN IP router needs to examine the IP Address to determine how to route a packet. It needs to determine which MAC address is associated with the IP Address in order to send the data correctly across the LAN. Once the packet arrives at the Switch, the Switch needs to determine if there is a VLAN ID that it can send the message to. Even if a node is on the same LAN Segment or even sharing the same LAN port, it cannot accept a message if it is destined to a VLAN to which it does not belong. Assigning one VLAN ID to a set of nodes and a different VLAN ID to another set of nodes isolates the two sets of nodes so that they cannot intercept messages from each other over the LAN. Thus, assigning VLAN IDs provides a measure of security between the two types of nodes. VLAN security has been called into question in recent years because a VLAN ID and a MAC address can be spoofed (although not with OSX OSAs in the zEnterprise)..



1. At the bottom of this visual you see the format of a frame that carries no VLAN tag information and one that carries VLAN tag information.
2. At the top of the diagram you see that we have two IP stacks that are sharing the OSA.
3. Both stacks belong to the same LAN segment: 192.168.1.0/24. The OSA is connected to the switch in Access Mode, because no VLAN tag is being added by the switch when it sends data to the mainframe.
4. The switch sends untagged data to the OSA, which then sends untagged packets to z/OS or z/VM.
5. Note how the other end of the switch is configured in Trunk Mode. The switch adds a Tag to identify different VLANs and IP segments that are further out in the network.

Trunk Mode to the z Platform



Inbound to z/OS or z/VM:
- *Untagged packets arrive.*

Outbound from Switch:
- *Switch adds VLAN ID*

Stack OUTBOUND:
- *Stack registers VLAN ID with OSA port*
- *OSA adds VLAN tag to outbound packets*

Stack INBOUND:
- *OSA filters inbound traffic based on VLAN ID.*
- *VLAN membership enforced.*

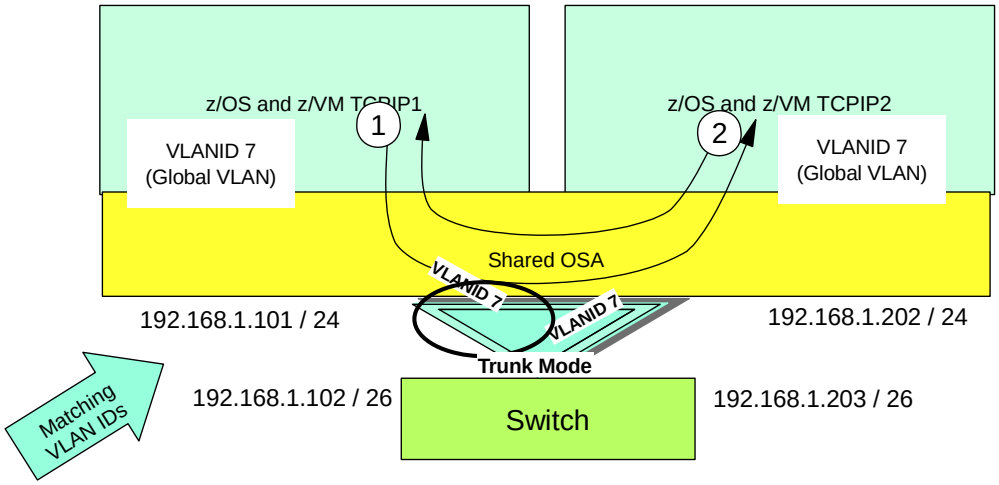
Untagged Ethernet frame

Destination MacAddress	Source MacAddress	Type / Length

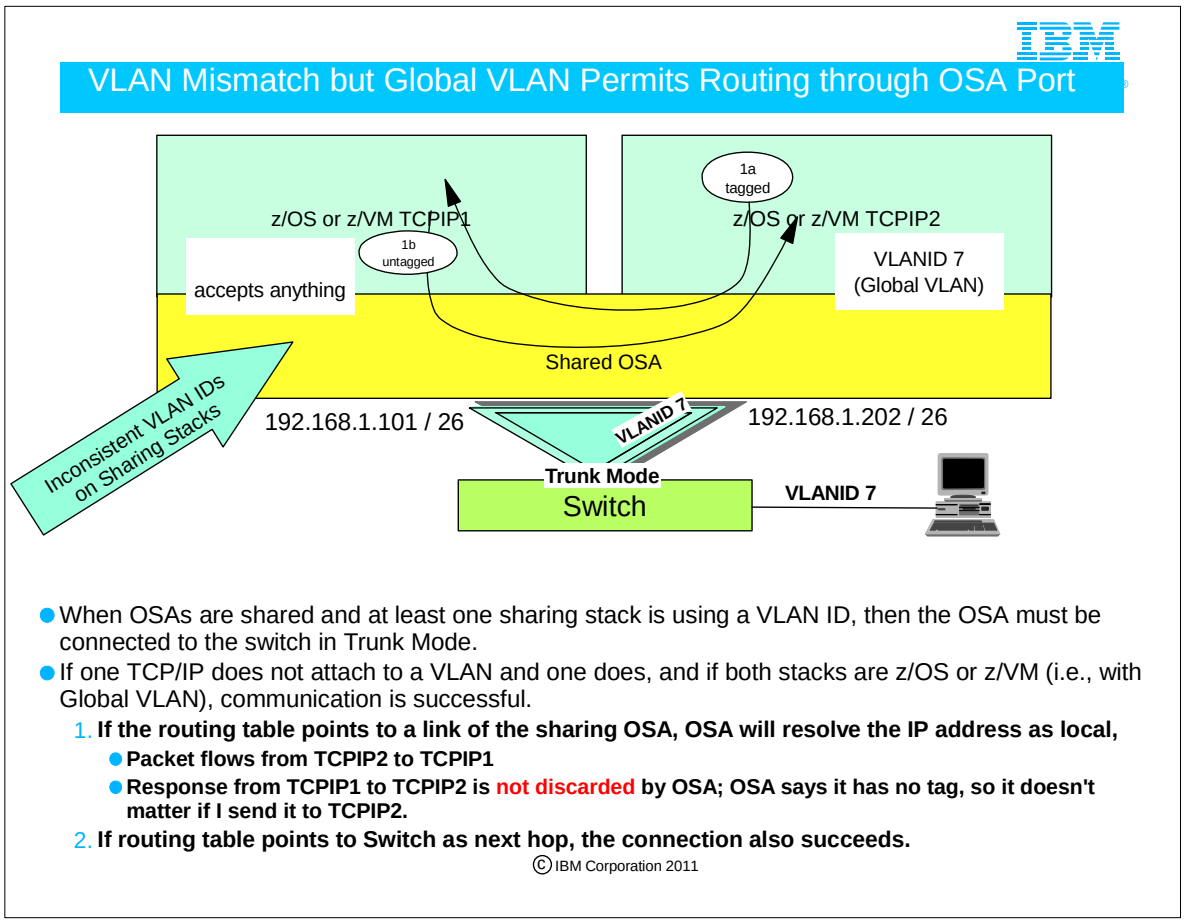
Tagged Ethernet frame : 802.1p or 802.1q

Destination MacAddress	Source MacAddress	Tag Control Information		Type / Length
		VLAN Tag 0x8100 (2 bytes)	Priority (3 bits) Canonical (1 bit) VLAN Identifier (12 bits)	

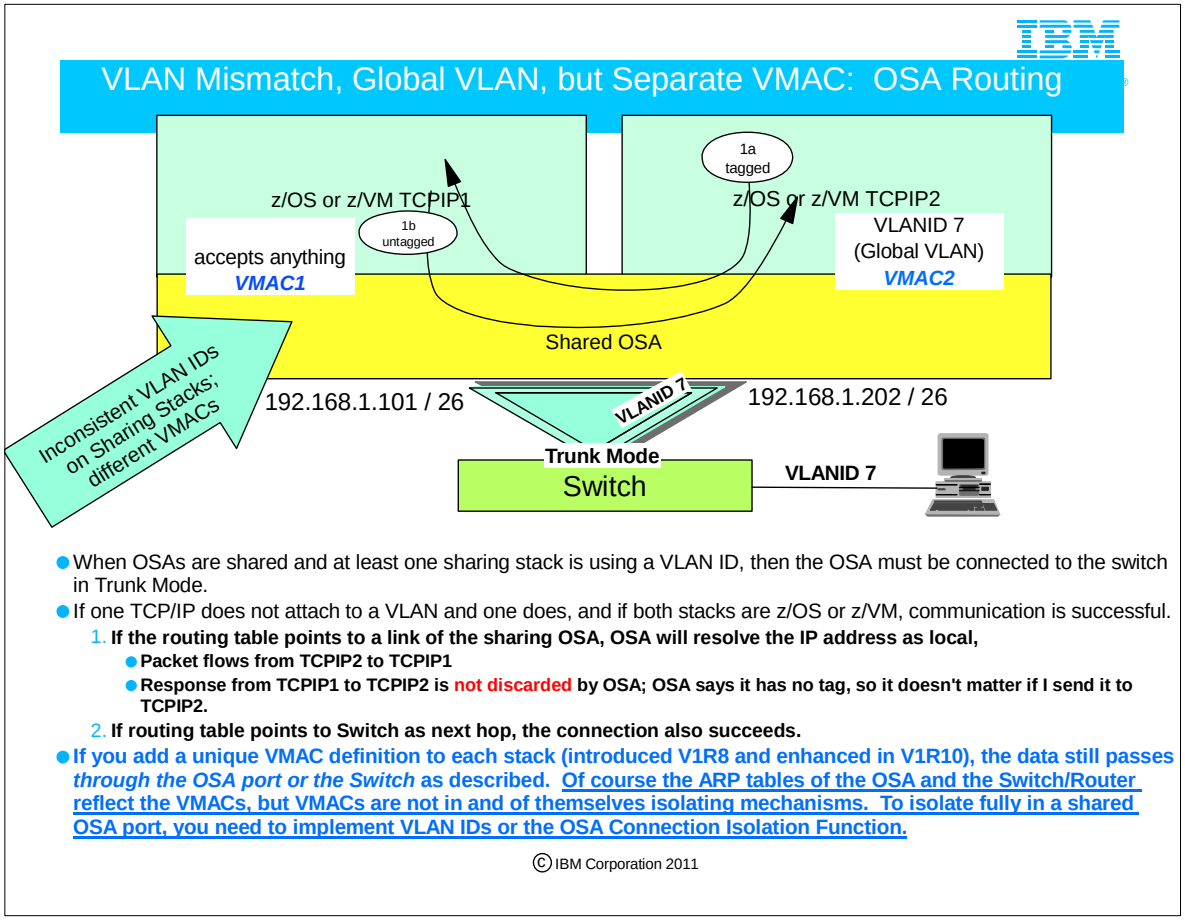
Remember VLAN ID Restrictions!
Shared VLAN & Routing through OSA Port



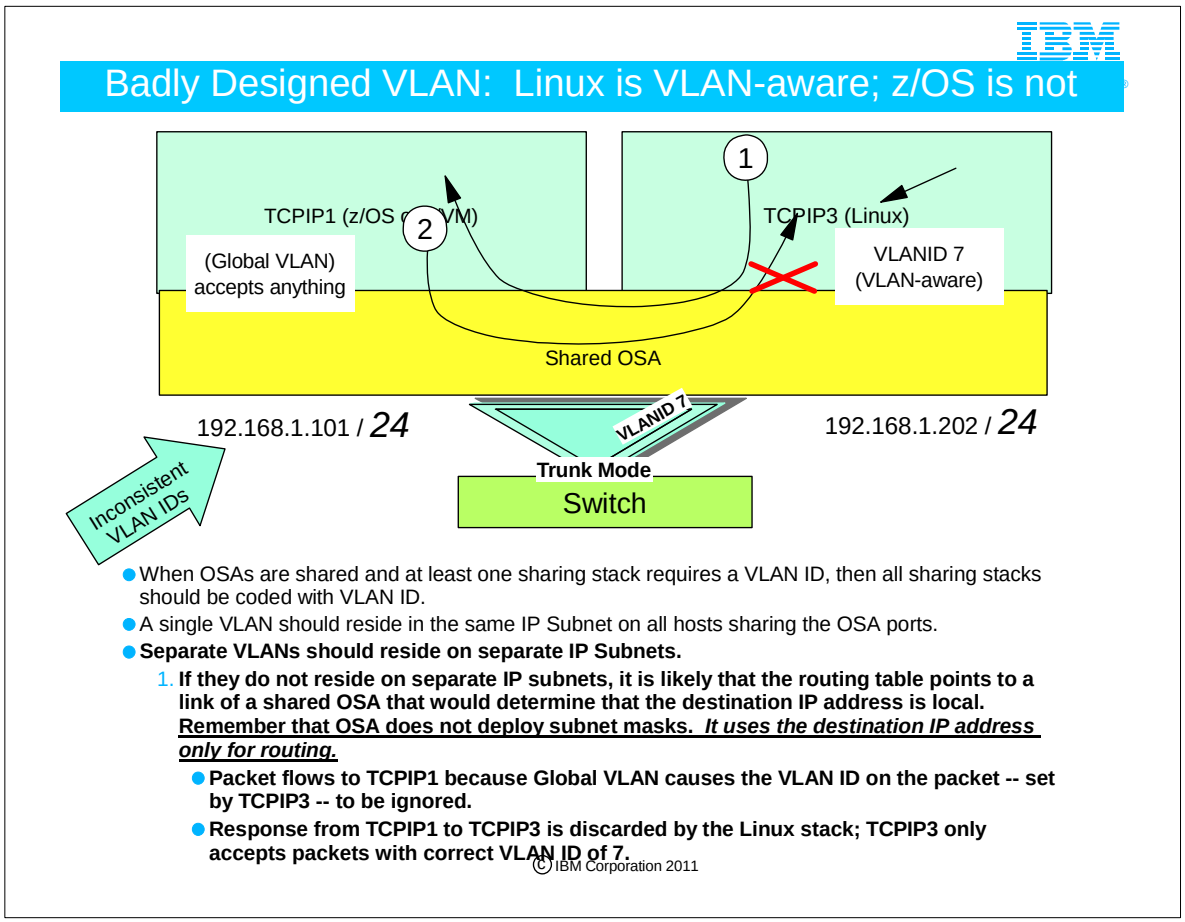
- When OSAs are shared and at least one sharing stack is using a VLAN ID, then the OSA must be connected to the switch in Trunk Mode.
- Best Practice: Each unique VLAN ID is on a separate IP (sub)net.
- **Traffic outbound from the IP Stack on z/OS or z/VM:**
  1. **If the routing table points to a link of the sharing OSA, OSA will resolve the VLAN ID and IP address as local and internally route to the target that is sharing the connection.**



1. In this diagram you seen that two TCP/IP stacks are sharing the OSA port. Both of the TCP/IP stacks support Global VLAN, because the stacks are either z/OS or z/VM or a combination of the two.
2. However, TCPIP1 has not been configured with a VLAN ID, whereas TCPIP2 has assigned VLAN ID of 7 to the OSA connection from that stack.
3. When an OSA port is shared and at least one of the sharing stacks is using a VLAN ID, then the OSA port must be connected to the switch in Trunk Mode.
4. Note how the inconsistent VLAN tagging in this scenario involving z/OS and z/VM presents no problem to communication across the internal OSA path.
5. As you see, TCPIP1 receives untagged packets because the OSA strips the tag in 1a, since the OSA knows that a Global VLAN stack does not care about the VLAN ID.
6. As long as the IP address in the packet matches the IP address in the stack (or as long as TCPIP1 is the PRIRouter or SECRouter for the received packet), the OSA delivers the packet to TCPIP1.



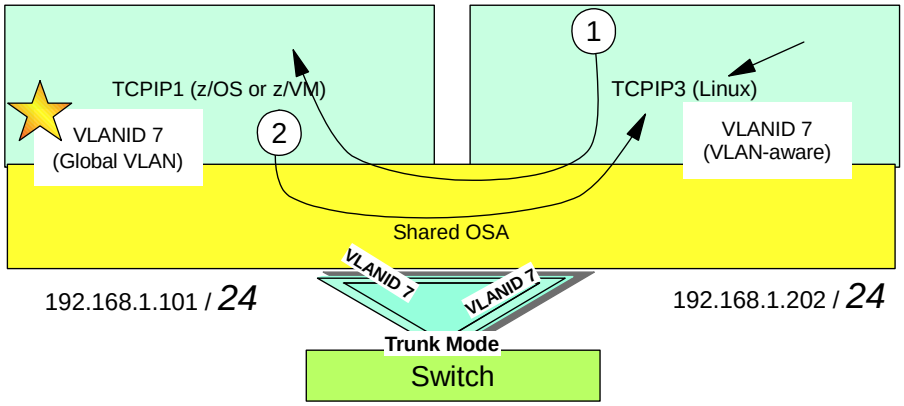
1. In this diagram you seen that two TCP/IP stacks are sharing the OSA port. Both of the TCP/IP stacks support Global VLAN, because the stacks are either z/OS or z/VM or a combination of the two.
2. However, TCPIP1 has not been configured with a VLAN ID, whereas TCPIP2 has assigned VLAN ID of 7 to the OSA connection from that stack.
3. When an OSA port is shared and at least one of the sharing stacks is using a VLAN ID, then the OSA port must be connected to the switch in Trunk Mode.
4. Note how the inconsistent VLAN tagging in this scenario involving z/OS and z/VM presents no problem to communication across the internal OSA path.
5. As you see, TCPIP1 receives untagged packets because the OSA strips the tag in 1a, since the OSA knows that a Global VLAN stack does not care about the VLAN ID.
6. As long as the IP address in the packet matches the IP address in the stack (or as long as TCPIP1 is the PRIRouter or SECRouter for the received packet), the OSA delivers the packet to TCPIP1.
7. If we add VMACs to the configuration as depicted, then the traffic still flows as before. However, PRIRouter or SECRouter coding is no longer necessary or even used. The Layer 3 routing table determines which MAC to send the data to.
8. If you add a unique VMAC definition to each stack (introduced V1R8 and enhanced in V1R10), the data still passes through the OSA port or the Switch as described. Of course the ARP tables of the OSA and the Switch/Router reflect the VMACs, but VMACs are not in and of themselves isolating mechanisms. To isolate fully in a shared OSA port, you need to implement VLAN IDs or the OSA Connection Isolation Function.



1. In this diagram you see that two TCP/IP stacks are sharing the OSA port. However one of the stacks uses the Global VLAN architecture, even though it has not established a VLAN ID on the OSA connection.
2. TCPIP3 is a VLAN-aware stack and it has assigned VLAN ID of 7 to the OSA connection from that stack.
3. When an OSA port is shared and at least one of the sharing stacks is using a VLAN ID, then the OSA port must be connected to the switch in Trunk Mode.
4. Note how the inconsistent VLAN tagging in this scenario involving z/OS (or z/VM) and Linux does present a problem for successful communication across the internal OSA path.
5. As you see, TCPIP1 receives untagged packets because the OSA strips the tag in 1, since the OSA knows that a Global VLAN stack does not care about the VLAN ID.
6. However, in (2), when TCPIP1 responds to the packet sent by Linux, the communication fails. The OSA delivers the untagged packet sent by TCPIP1 to TCPIP3, where Linux rejects it.

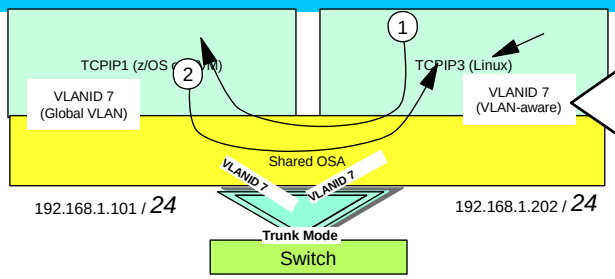
Solution to Badly Designed VLAN: Linux and z/OS or z/VM

- If one stack on a shared OSA port codes a VLAN ID, then all stacks should.
  - The z/OS or z/VM stack has now coded a VLANID on the Interface.



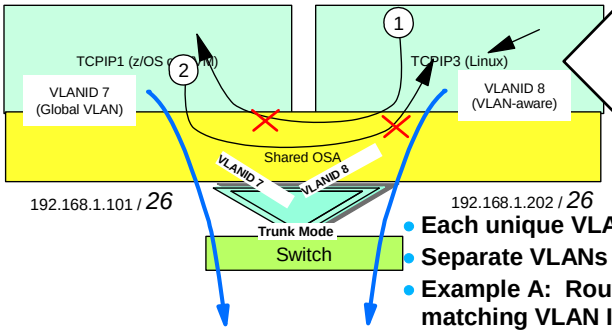
- When OSAs are shared, VLAN IDs should be set on all sharing TCP/IP stacks.
  1. It can be the same VLAN ID on each stack or a different one on each stack.
     - Each unique VLAN ID must be on a separate IP subnet.
- **Separate VLANs should reside on separate IP Subnets.**
  1. **If they do not reside on separate IP subnets, it is likely that the routing table points to a link of the sharing OSA. OSA will resolve the VLAN ID and IP address as local and internally route to the target that is sharing the connection.**
     - **This example: OSA routes correctly to either stack; VLAN IDs match in both directions.**

Correctly Designing VLANs



A

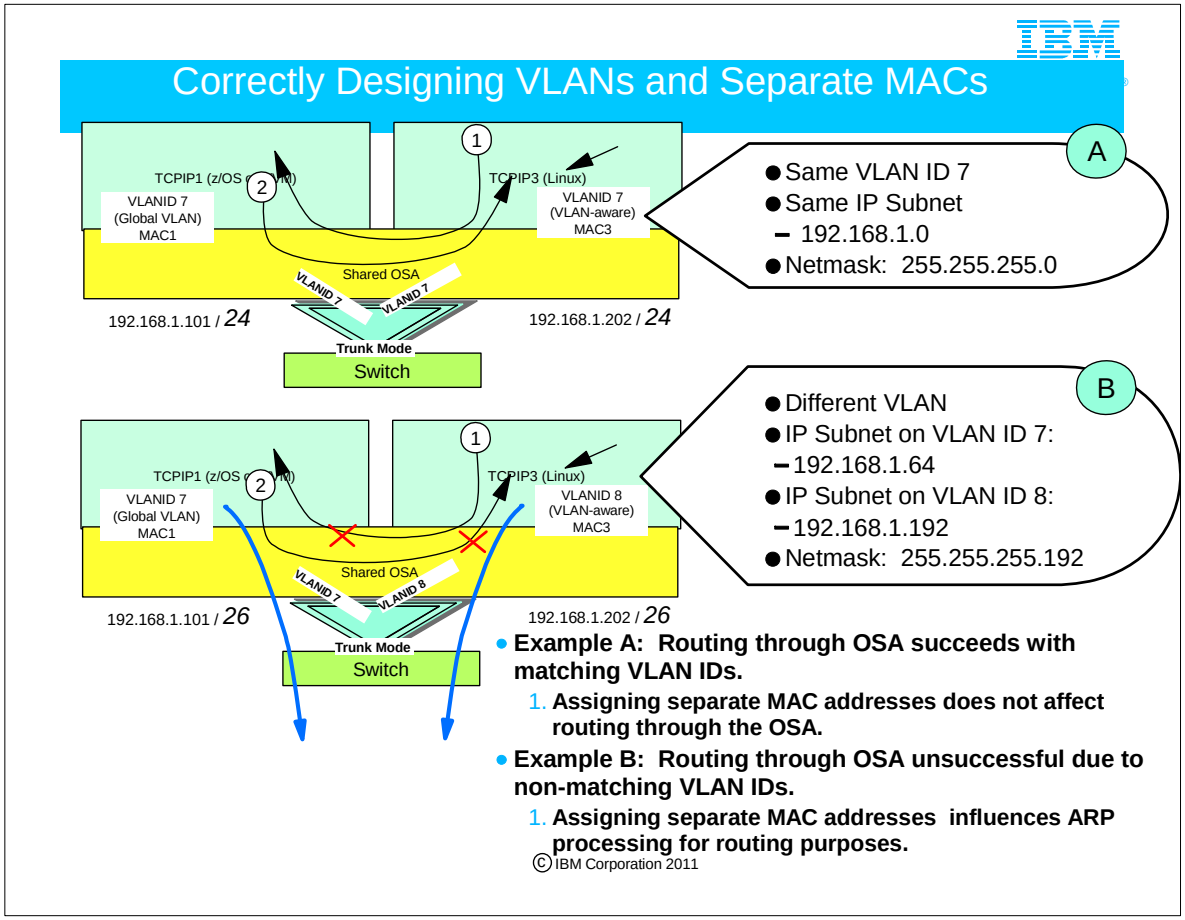
- Same VLAN ID 7
- Same IP Subnet
  - 192.168.1.0
- Netmask: 255.255.255.0



B

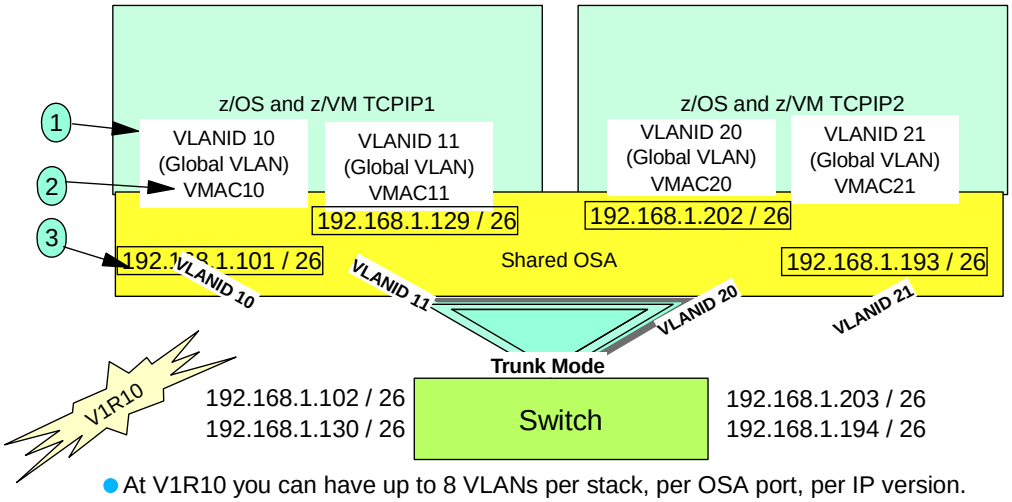
- Different VLAN
- IP Subnet on VLAN ID 7:
  - 192.168.1.64
- IP Subnet on VLAN ID 8:
  - 192.168.1.192
- Netmask: 255.255.255.192

- **Each unique VLAN ID should reside in its own IP Subnet.**
- **Separate VLANs should reside on separate IP Subnets.**
- **Example A: Routing through OSA succeeds with matching VLAN IDs.**
- **Example B: Routing through OSA unsuccessful due to non-matching VLAN IDs. (VLANs can segment/segregate!)**



1. Note how Target MAC plays no role when routing through an OSA (i.e., across an OSA port).

Multiple VLAN Support Starting in z/OS CS V1R10 Shared OSA

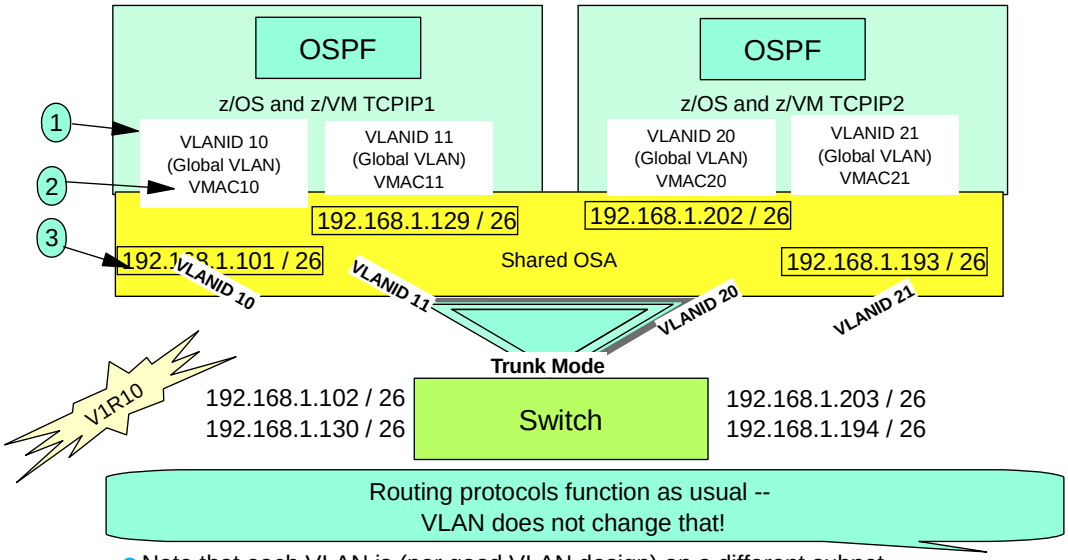


- At V1R10 you can have up to 8 VLANs per stack, per OSA port, per IP version.
  1. With multiple VLAN IDs per stack on an OSA port, you must assign a VLAN ID to every one of the multiple Interfaces on that OSA port and
  2. You must assign or default to separate VMACs on each VLAN ID.
     - **The separate VMACs make each interface look like a DEDICATED OSA Port.**
  3. As usual, each VLAN ID must be on a separate subnet.
- **Add OSPF to this picture and ...**

OSPF and VLANs; Tuning for Multiple VLAN IDs

- Use Virtual LANs to segment a Physical LAN port into multiple Logical LAN Ports.

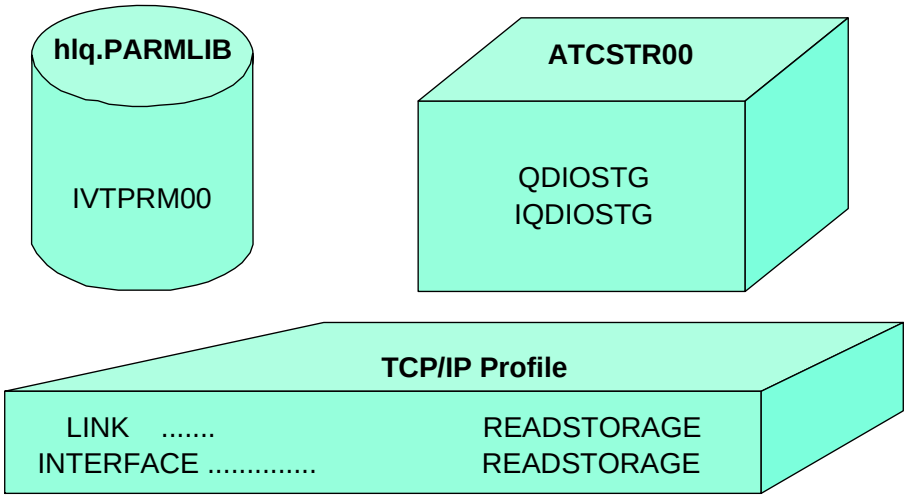
Multiple VLAN Support in z/OS CS V1R10
Shared OSA & Routing through OSA Port



- Note that each VLAN is (per good VLAN design) on a different subnet.
- Therefore, OSPF, though not aware of VLAN IDs, is aware of the separate subnets and OSPF manages this as usual: separate Interfaces on separate IP Subnets with Link State Advertisements flowing over each.

Tuning Storage with VLAN IDs

- **QDIO (and HiperSockets) use fixed storage for READ processing.**
  - **Multiple VLANs on an OSA Port may require adjustments for:**



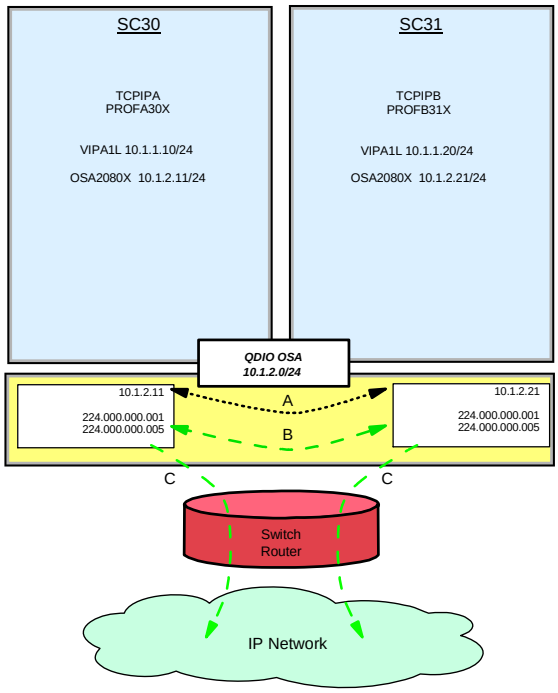
- Monitor with RMF and with VTAM TNSTATS

1. Each OSA-Express QDIO and HiperSockets interface requires fixed storage for read processing (which is allocated by VTAM). If you define a large number of these interfaces (for example, by configuring multiple VLANs to one or more OSA-Express features), then you need to consider how much fixed storage your configuration requires.
   1. For example, each active OSA-Express QDIO DATAPATH device consumes a large amount of fixed storage. Defining a large number (for example, 8 or more devices per z/OS image) of QDIO devices can cause z/OS Communications Server to consume a significant amount of fixed storage. This could lead to degradation of overall system performance. When configuring a large number of devices, it is important to use the controls provided to manage and tune the amount of fixed storage consumed by these devices. Review the following parameters with this in mind:
      1. – VTAM QDIOSTG start option
      2. – READSTORAGE specifications in the TCP/IP profile
      3. – FIXED MAX specification in the IVTPRM00 parmlib member for Communication Storage Manager (CSM
2. For information about how much fixed storage VTAM allocates by default for each OSA-Express QDIO and HiperSockets interface, how to control the amount of this storage allocation using the VTAM QDIOSTG start option (for OSA-Express QDIO) and the VTAM IQDIOSTG start option (for HiperSockets), and considerations for the IVTPRM00 parmlib member, see z/OS Communications Server: SNA Resource Definition Reference.
3. You can also override the global QDIOSTG or IQDIOSTG value and control the amount of fixed storage for a specific OSA-Express QDIO or HiperSockets interface by using the READSTORAGE parameter on the LINK and INTERFACE statements.
4. QDIOSTG:
   1. Specifies how much storage VTAM keeps available for read processing for all OSA QDIO data devices. Units are defined in QDIO SBALs (QDIO read buffers). Each SBAL is 64k. For most users the default setting will be the most suitable option. The storage used for this read processing is allocated from CSM data space 4k pool, and is fixed storage. The IBM recommended values can be configured by specifying MAX, AVG, or MIN, which are predefined constants (number of SBALs) that are most appropriate for this type of adapter.
   2. You can use VTAM tuning stats to evaluate your needs and usage. Under a sample (typical) workload, the NOREADS counter should remain low (close to 0). If this count does not remain low you might need to consider a higher setting for QDIOSTG.
5. IQDIOSTG:
   1. Specifies how much storage VTAM keeps available for read processing for all HiperSockets data devices that use a MFS (Maximum Frame Size) of 64k. The HiperSockets MFS is defined in HCD. The HiperSockets storage units are defined in QDIO SBALs (QDIO read buffers). Each SBAL is 64k. For most users, the default setting will be the most suitable option. The storage used for this read processing is allocated from CSM data space 4k pool, and is fixed storage. HiperSockets devices that are defined with a smaller MFS (16k, 24k, or 40k) are not affected by this start option. Those devices will use 126 SBALs.
   2. You can use VTAM tuning stats to evaluate your needs and usage. Under a sample (typical) workload, the NOREADS counter should remain low (close to 0). If this count does not remain low you might need to consider a higher setting for IQDIOSTG. RMF can also be used to evaluate the correct setting for your environment. RMF records send failures, which can be an indication that the target LP (logical partition) does not have enough storage (read SBALs). 3. You can override the IQDIOSTG value for a given HiperSockets device by using the READSTORAGE parameter on the IPAQIDIO LINK statement or the IPAQIDIO6 INTERFACE statement on the TCP/IP profile.
6. READSTORAGE
   1. An optional parameter indicating the amount of fixed storage that z/OS Communications Server should keep available for read processing for this adapter. The QDIOSTG VTAM start option allows you to specify a value which applies to all OSA-Express adapters in QDIO mode. You can use the READSTORAGE keyword to override the global QDIOSTG value for this adapter based on the inbound workload you expect over this adapter on this stack. The default value is GLOBAL, which stands for the QDIOSTG VTAM start option.

OSA Connection Isolation

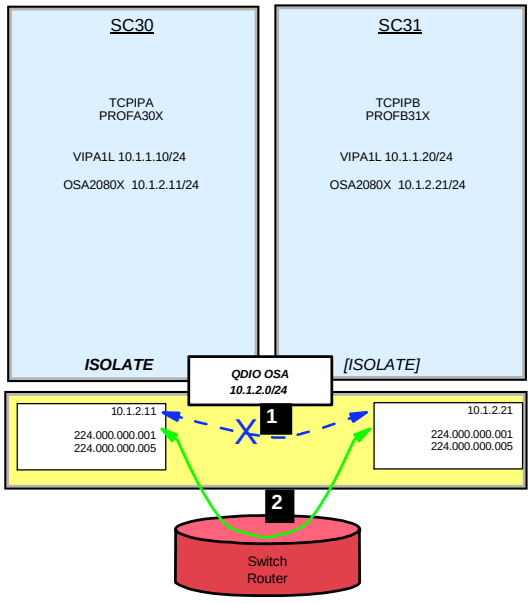
- **Carefully implement Connection Isolation in z/VM and z/OS to eliminate the internal path through the OSA port.**

Routing of Traffic in a Shared OSA Port



1. Another method available to isolate traffic across a shared OSA port is OSA Connection Isolation. This method can be deployed with or without out assigning a VLAN ID or a VMAC to the OSA port.
2. Many customers share OSA-Express ports across logical partitions, especially if capacity is not an issue. Each stack sharing the OSA port registers certain IP addresses and multicast groups with the OSA.
3. For performance reasons, the OSA-Express bypasses the LAN and routes packets directly between the stacks when possible.
4. For unicast packets, OSA internally routes the packet when the next-hop IP address is registered on the same LAN or VLAN by another stack sharing the OSA port.
   1. A: You see how TCPIPA routes a packet to 10.1.2.21 in TCPIPB over the OSA port without exiting out onto the LAN because the next hop to reach the destination is registered in the OSA Address Table (OAT); the TCPIPA routing table indicates that the destination can be reached by hopping through the direct connection to the 10.1.2.0/24 network.
   2. B For multicast (e.g., OSPF protocol packets), OSA internally routes the packet to all sharing stacks on the same LAN or VLAN which registered the multicast group. Note how TCPIPA and TCPIPB have each registered multicast addresses for OSP (224.000.000.00n) in the OSA port.
   3. C OSA also sends the multicast/broadcast packet to the LAN. For broadcast (not depicted), OSA internally routes the packet to all sharing stacks on the same LAN or VLAN.
5. Some customers express concerns about this efficient communication path and wish to disable it; they may wish to disable the function because traffic flowing internally through the OSA adapter bypasses any security features implemented on the external LAN

OSA Connection Isolation (1.11)



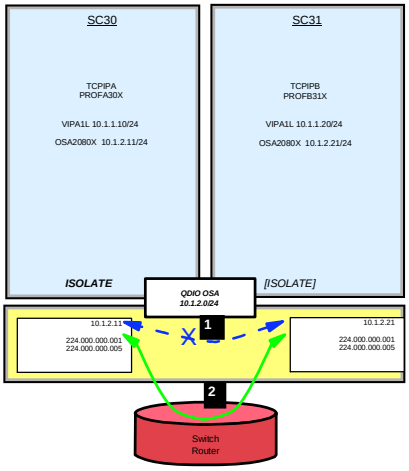
INTERFACE OSA2080X
DEFINE IPAQENET
PORTNAME OSA2080
IPADDR 10.1.2.11/24
MTU 1492
VLANID 10
VMAC ROUTEALL
ISOLATE

INTERFACE OSA2080X
DEFINE IPAQENET
PORTNAME OSA2080
IPADDR 10.1.2.21/24
MTU 1492
VLANID 10
VMAC ROUTEALL
[ISOLATE]

1 = Direct route between TCPIPA and TCPIPB
2 = Indirect route between TCPIPA and TCPIPB

1. Some environments require strict controls for routing data traffic between servers or nodes. In certain cases, the LPAR-to-LPAR capability of a shared OSA port can prevent such controls from being enforced. For example, you may need to ensure that traffic flowing through the OSA adapter does not bypass firewalls or intrusion detection systems implemented on the external LAN. We have described several ways to isolate traffic from different LPARs on a shared OSA port, with one of these methods being OSA Connection Isolation.
2. The feature is called OSA Connection Isolation in z/OS, but it is also available in z/VM, where it is called QDIO data connection isolation or VSWITCH port isolation. It allows you to disable the internal routing on a QDIO connection basis, providing a means for creating security zones and preventing network traffic between the zones. It also provides extra assurance against a misconfiguration that might otherwise allow such traffic to flow as in the case of an incorrectly defined IP filter. With interface isolation, internal routing can be controlled on an LPAR basis. When interface isolation is enabled, the OSA will discard any packets destined for a z/OS LPAR that is registered in the OAT as isolated.
3. QDIO interface isolation is supported by Communications Server for z/OS V1R11 and all OSA-Express3 and OSA-Express2 features on System z10, and by all OSA-Express2 features on System z9, with an MCL update. Refer to the appropriate Preventive Service Planning bucket for details regarding your System z server.
4. Coding ISOLATE on your INTERFACE statement enables the function. It tells the OSA-Express not to allow communications to this stack other than over the LAN.
    1. As the visual depicts, the ISOLATE parameter is available only on the INTERFACE statement. To eliminate the direct path through the OSA between the two eepcited LPARs, you need code ISOLATE on only one of the two INTERFACEs. We have coded it on both in order to assure, that if any other LPAR starts sharing the OSA port, that other LPAR cannot use the direct path to communicate even with TCPIPB..
5. If you attempt to code ISOLATE on an INTERFACE that does not support the ISOLATE function, you receive a message:
    1. EZD0022I INTERFACE OSA2080X DOES NOT SUPPORT THE ISOLATE FUNCTION
6. Dynamic routing protocol implementations with RIP or OSPF require careful planning on LANs where OSA-Express connection isolation is in effect; the dynamic routing protocol learns of the existence of the direct path but is unaware of the isolated configuration, which renders the direct path across the OSA port to the registered target unusable. If the direct path that is operating as ISOLATEd is selected, you will experience routing failures.
7. If the visibility of such errors is undesirable, you can take other measures to avoid the failure messages. If you are simply attempting to bypass the direct route in favor of another, indirect route, you can accomplish this as well with some thoughtful design.
8. For example, you might purposely bypass the direct path by using Policy Based Routing (PBR) or by coding static routes that supersede the routes learned by the dynamic routing protocol. You might adjust the weights of connections to favor alternate interfaces over the interfaces that have been coded with ISOLATE.
9. If, however, TCPIPA and TCPIPB do need to exchange information, you will need to deploy an effective route that bypasses the direct route between them. Therefore, at TCPIPA you might add a non-replaceable static route to an IP address in TCPIPB; the static route in the BEGINROUTES block points to the next-hop router on the path indicated with (2) in the visual.

OSA Connection Isolation: Dynamic Routing Considerations (1.11)



1 = Direct route between TCPIPA and TCPIPB
2 = Indirect route between TCPIPA and TCPIPB

- Combine OMPROUTE with Static Routes to bypass direct routing through OSA port.

```
;TCPIPA.TCPPARMS(ROUTA30X)
; AUTOLOG LIST:  INITIALIZE OMPROUTE
...
BEGINRoutes
; Direct Routes - Routes directly connected to my interfaces
; Destination  Subnet Mask   First Hop     Link Name   Packet Size
 ROUTE 10.1.2.0/24          10.1.2.240    OSA2080X    mtu 1492
 ROUTE 10.1.1.0/24          10.1.2.240    OSA2080X    mtu 1492
 ROUTE 10.1.1.20/32         10.1.2.240    OSA2080X    mtu 1492
 ENDRoutes
```

```
;TCPIPB.TCPPARMS(ROUTB31X)
; AUTOLOG LIST:  INITIALIZE OMPROUTE
...
BEGINRoutes
; Direct Routes - Routes directly connected to my interfaces
; Destination  Subnet Mask   First Hop     Link Name   Packet Size
;
 ROUTE 10.1.2.0/24          10.1.2.240    OSA2080X    mtu 1492
 ROUTE 10.1.1.0/24          10.1.2.240    OSA2080X    mtu 1492
 ROUTE 10.1.1.10/32         10.1.2.240    OSA2080X    mtu 1492
 ENDRoutes
```

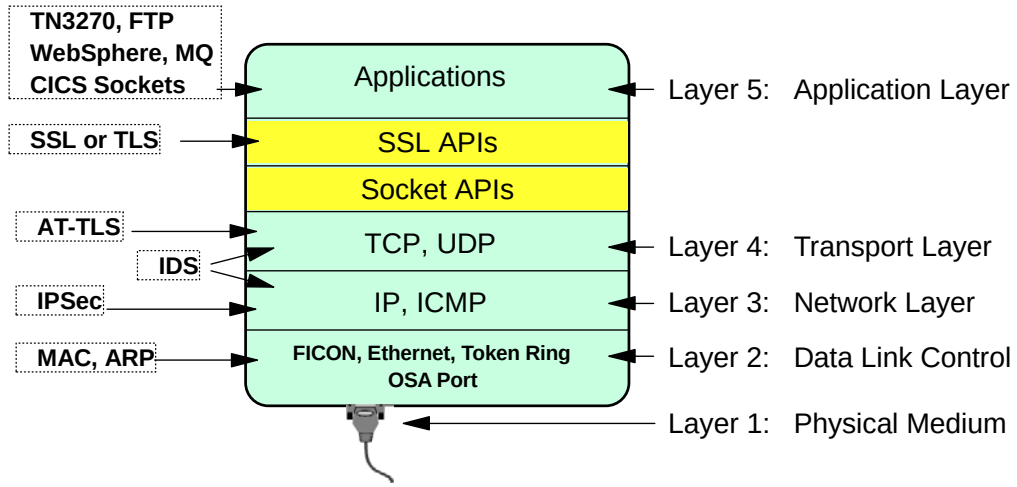

1. Some environments require strict controls for routing data traffic between servers or nodes. In certain cases, the LPAR-to-LPAR capability of a shared OSA port can prevent such controls from being enforced. For example, you may need to ensure that traffic flowing through the OSA adapter does not bypass firewalls or intrusion detection systems implemented on the external LAN. We have described several ways to isolate traffic from different LPARs on a shared OSA port, with one of these methods being OSA Connection Isolation.
2. The feature is called OSA Connection Isolation in z/OS, but it is also available in z/VM, where it is called QDIO data connection isolation or VSWITCH port isolation. It allows you to disable the internal routing on a QDIO connection basis, providing a means for creating security zones and preventing network traffic between the zones. It also provides extra assurance against a misconfiguration that might otherwise allow such traffic to flow as in the case of an incorrectly defined IP filter. With interface isolation, internal routing can be controlled on an LPAR basis. When interface isolation is enabled, the OSA will discard any packets destined for a z/OS LPAR that is registered in the OAT as isolated.
3. QDIO interface isolation is supported by Communications Server for z/OS V1R11 and all OSA-Express3 and OSA-Express2 features on System z10, and by all OSA-Express2 features on System z9, with an MCL update. Refer to the appropriate Preventive Service Planning bucket for details regarding your System z server.
4. Coding ISOLATE on your INTERFACE statement enables the function. It tells the OSA-Express not to allow communications to this stack other than over the LAN.
    1. As the visual depicts, the ISOLATE parameter is available only on the INTERFACE statement. To eliminate the direct path through the OSA between the two eepcited LPARs, you need code ISOLATE on only one of the two INTERFACEs. We have coded it on both in order to assure, that if any other LPAR starts sharing the OSA port, that other LPAR cannot use the direct path to communicate even with TCPIPB..
5. If you attempt to code ISOLATE on an INTERFACE that does not support the ISOLATE function, you receive a message:
    1. EZD0022I INTERFACE OSA2080X DOES NOT SUPPORT THE ISOLATE FUNCTION
6. Dynamic routing protocol implementations with RIP or OSPF require careful planning on LANs where OSA-Express connection isolation is in effect; the dynamic routing protocol learns of the existence of the direct path but is unaware of the isolated configuration, which renders the direct path across the OSA port to the registered target unusable. If the direct path that is operating as ISOLATEd is selected, you will experience routing failures.
7. If the visibility of such errors is undesirable, you can take other measures to avoid the failure messages. If you are simply attempting to bypass the direct route in favor of another, indirect route, you can accomplish this as well with some thoughtful design.
8. For example, you might purposely bypass the direct path by using Policy Based Routing (PBR) or by coding static routes that supersede the routes learned by the dynamic routing protocol. You might adjust the weights of connections to favor alternate interfaces over the interfaces that have been coded with ISOLATE.
9. If, however, TCPIPA and TCPIPB do need to exchange information, you will need to deploy an effective route that bypasses the direct route between them. Therefore, at TCPIPA you might add a non-replaceable static route to an IP address in TCPIPB; the static route in the BEGINROUTES block points to the next-hop router on the path indicated with (2) in the visual.
10. The effect of ICMP redirect packets: To avoid the override of the ICMP redirect packets that would most likely occur from the router to the originating host, you need to disable the receipt of ICMP redirects in the IP stacks or disable ICMP redirects at the router. If you are using OMPROUTE, ICMP redirects are automatically disabled, as evidenced by the message that appears during OMPROUTE initialization:
    1. EZZ7475I ICMP WILL IGNORE REDIRECTS DUE TO ROUTING APPLICATION BEING ACTIVE
11. The visual shows the coding for Static non-replaceable routes at TCPIPA and TCPIPB to override direct route through OSA port

OSA Connection Isolation (1.11)

```
**************************************************************************
*** OSA/SF Get OAT output created 10:46:14 on 09/23/2009               ***
*** IOACMD APAR level - OA26486                                        ***
*** Host   APAR level - OA26486                                        ***
**************************************************************************
***              Start of OSA address table for CHPID 02              ***
**************************************************************************
* UA(Dev) Mode     Port     Entry specific information      Entry  Valid
**************************************************************************
                              Image 2.3 (A23     ) CULA 0
80(2080)* MPC       N/A     OSA2080  (QDIO control)              SIU ALL
82(2082)  MPC  00 No4  No6  OSA2080  (QDIO data)    Isolated    SIU ALL
                 VLAN   10   (IPv4)

                 Group Address   Multicast Address
                 01005E000001    224.000.000.001
                 01005E000005    224.000.000.005

                 VMAC            IP address
          HOME   020010749925    010.001.002.011

83(2083)  MPC  00 No4  No6  OSA2080  (QDIO data)                S   ALL

...
```


1. Even the OSA/SF display shows where ISOLATE is enabled, as you can see from the display.

Security Mechanisms for Segmentation



- Encryption can be considered a form of segmentation.
- Consider that encrypted traffic that flows through an OSA port is unintelligible even to OSAENTA (a trace) and is certainly isolated from other traffic flowing through the same shared OSA port.
  - IPSec and SSL/TLS/AT-TLS provide
    - encryption, authentication, data integrity checking

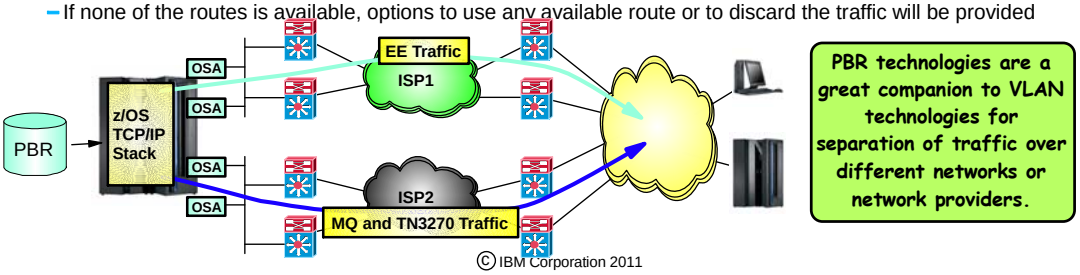
1. Each level of any security architecture can implement security functions.
2. For example, certain applications have implemented their own security by enabling calls to RACF, by enabling Access Control Lists, by specifying which data sets MUST be used (thus excluding other data sets), and so on.
3. For example, many of the other layers of the TCP/IP stck implement the types of sec urity that Policy Agent policies can exploit: SSL/TLS (AT-TLS), IDS, IPSec, IP Filtering, etc.
4. For example, it is possible at the DLC layer to invoke MAC filtering for the purposes of security.
5. One should not think of security only in terms of encryption -- there are many layers of security that can be implemented.
6. PHYSICAL: This is the actual hardware ... adapter and cabling ... that connects the TCP/IP node with the external network.
7. LINK: Layer 2 is the data link layer. This layer is also called simply the link layer. The actual protocols encompassed in the link layer are numerous, and the implementation details can be found in various documents throughout the Internet and in trade texts. The foremost data link layer protocol is the Ethernet protocol.
8. MAC: Ethernet designates the frame format and the speed of the data travelling over the physical network. However, there is still a need for controlling how individual hosts (workstations) attached to the physical network locate each other. The answer is the media access control (MAC) address. Every host connected to the network has a unique MAC address associated with its NIC. This MAC address, via the NIC, uniquely identifies the host.
9. ARP: The Address Resolution Protocol is a layer 2 protocol used to map MAC addresses to IP addresses. All hosts on a network are located by their IP address, but NICs do not have IP addresses, they have MAC addresses. ARP is the protocol used to associate the IP address to a MAC address.
10. IP: The most significant protocol at layer 3 (also called the network layer) is the Internet Protocol, or IP. IP is the standard for routing packets across interconnected networks—hence, the name internet. It is an encapsulating protocol. The format of an IP packet is documented in RFC 791. The most significant aspect of the IP protocol is the addressing: every IP packet includes the IP source address (where the packet is coming from) and the IP destination address (where the packet is heading to).
11. ICMP: ICMP is actually a user of the IP protocol—in other words, ICMP messages must be encapsulated within IP packets. However, ICMP is implemented as part of the IP layer. So ICMP processing can be viewed as occurring parallel to, or as part of, IP processing.
12. TRANSPORT LAYER: This layer deals with the actual delivery of a packet to an application. Two protocols are available at Layer 4: TCP and UDP. TCP and UDP know about the port numbers of applications that they are to deliver data to.
13. TCP: TCP is always referred to as a connection-oriented protocol. What this entails is that prior to any communication occurring between two endpoints, a connection must be established. During the communications (which can last for seconds or for days) the state of the connection is continually tracked. And, when the connection is no longer needed, the connection must be ended.
14. UDP: A UDP header containing an IP address and a port number is wrapped around whatever data needs to be sent, and the packet is handed over to the IP layer. As long as the lower layers do their jobs correctly, the remote end should receive the datagram as expected. There are no acknowledgement counters and no connection states.
15. SOCKETS: The term socket in a TCP or UDP context fully describes the endpoint of a connection. The socket is consequently a combination of an IP address, a port number, and the protocol being used.
16. APPLICATIONS: Sitting above layer 4 are the applications. Applications are recognized by their port numbers. Port numbers are TCP's method of knowing which application should receive a packet. Applications can use either TCP or UDP to communicate. Because of its inherent reliability, TCP tends to be used more often. Examples of applications running on z/OS using TCP include sendmail, Web servers, FTP and telnet. Applications using UDP on z/OS are Traceroute, Enterprise Extender, and name servers (Domain Name System).
17. OSAENTA: To assist in problem diagnosis, the OSA-Express network traffic analyzer (OSAENTA) function provides a way to trace inbound and outbound frames for an OSA-Express2 feature in QDIO mode. The OSAENTA trace function is controlled and formatted by z/OS Communications Server, but is collected in the OSA at the network port. You can control the OSAENTA trace function using either the OSAENTA statement in the TCP/IP profile or the VARY TCPIP,,OSAENTA command.
18. Because the data is collected at the Ethernet frame level, this function enables you to trace the MAC headers for packets, a capability not provided by existing packet traces. It also enables the tracing of other types of packets that existing packet traces do not contain, including the following:
    1. ARP packets
    2. Packets to and from other users sharing the OSA, including other TCP/IP stacks, z/Linux users, and z/VM users
    3. SNA packets
19. Security for OSAENTA is controlled through HMC functions and SAF-provided access to execute the OSAENTA functions..

OSPF and PBR

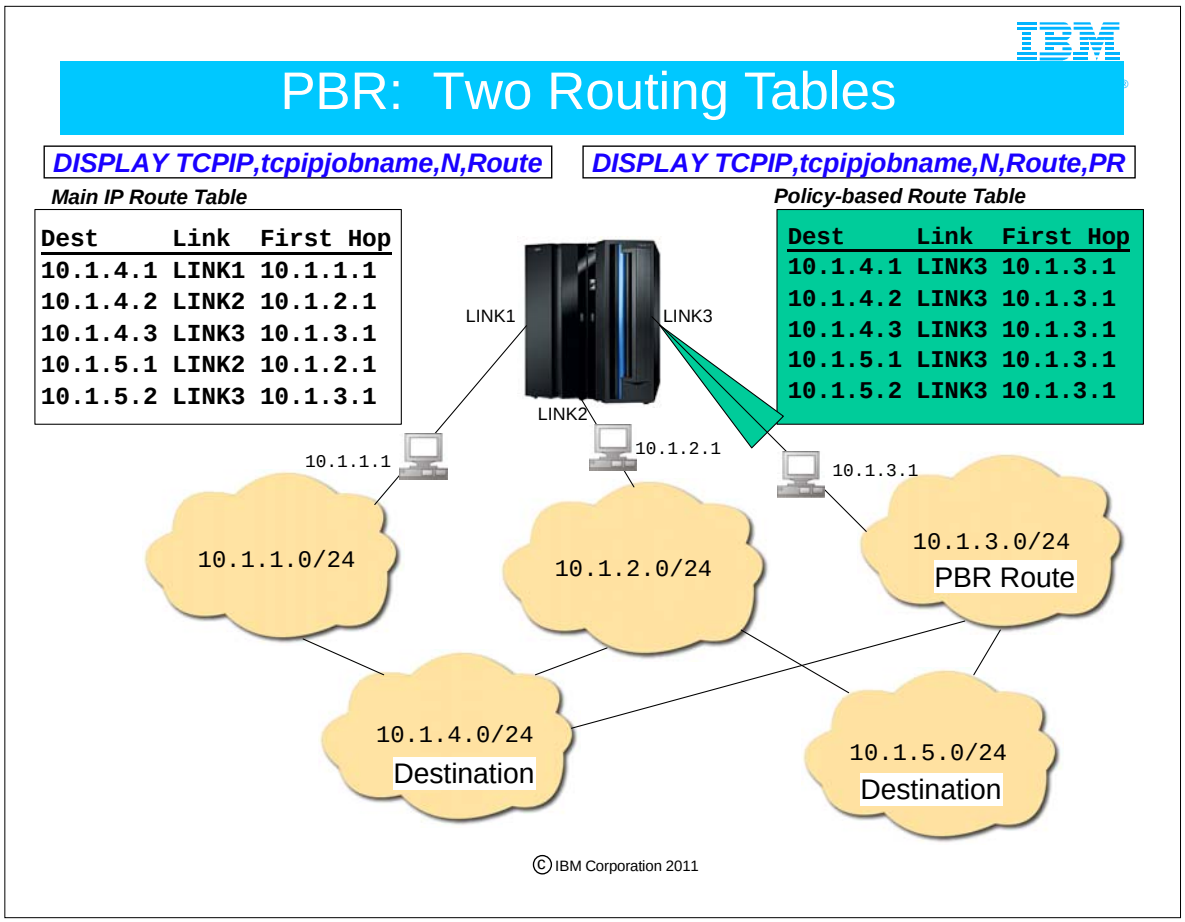
- Use Policy Based Routing at z/OS and in the External Routers
  - to isolate available routing paths according to security

Policy-based Outbound Routing:Segment the Traffic Originating on z/OS

- **What does Policy Based Routing (PBR) do?**
  - Choose first hop router, outbound network interface (including VLAN), and MTU
    - Can be defined with static routes and/or dynamic routes
  - Choice can be based on more than the usual destination IP address/subnet
    - With PBR, the choice can be based on source/destination IP addresses, source/destination ports, TCP/UDP, JOB/APPLname, Security Label, NetAccess Security Zone
- **Allows an installation to separate outbound traffic for specific applications to specific network interfaces and first-hop routers:**
  - Security related
  - Choice of network provider
  - Isolation of certain applications
    - EE traffic over one interface
    - TN3270 traffic over another interface
  - PBR policies will identify one or more routes to use
    - If none of the routes is available, options to use any available route or to discard the traffic will be provided



1. PBR can be used to segregate tarffic across different interfaces. PBR must also be implemented in the routers so that the reverse path is also correctly infuenced.



1. In this sample, we have a node connected to a set of IP subnets. You can see, in the partial table shown, that the main route table contains routes to destinations throughout the network and that these routes use all of the three available network links. These may be routes that were added to the main route table by OMPROUTE, in which case the location of the destinations and the dynamic routing configuration throughout the network has resulted in these routes being the best routes available. If there is a need for a certain type of IP traffic (for example all traffic sent by a specific job name) to be sent out LINK3, a policy-based route table such as the one shown could be created. In this particular case, the policy-based route table contains routes to all of the same destinations as are in the main route table. However, all of the routes in the policy-based route table use LINK3.
2. Once the set of route tables that can be used for some type of outbound traffic has been determined, how does IP Routing search for a route in those tables?
3. Most often there will be one policy-based route table defined to be used for the traffic, but there may be as many as eight. Each of the policy-based route tables is searched, in the order defined, for a route to the destination. If any active route to the destination is found in a route table, the search is stopped and that route is used for the traffic. This route may be a host route, a subnet, network, or supernet route, or a default route. If no active route to the destination is found in a route table, the search continues with the next route table. If all policy-based route tables are searched without success, the main route table may also be searched if the policy indicates that the main route table can be used as a backup.
4. OMPROUTE learns about the policy-based route tables, and the parameters for controlling them, from the TCP/IP stack

PBR: OSPF Changes and Displays

DISPLAY TCPIP,tcpipjobname,OMProute,RTTABLE,PRtable=SECLOW2

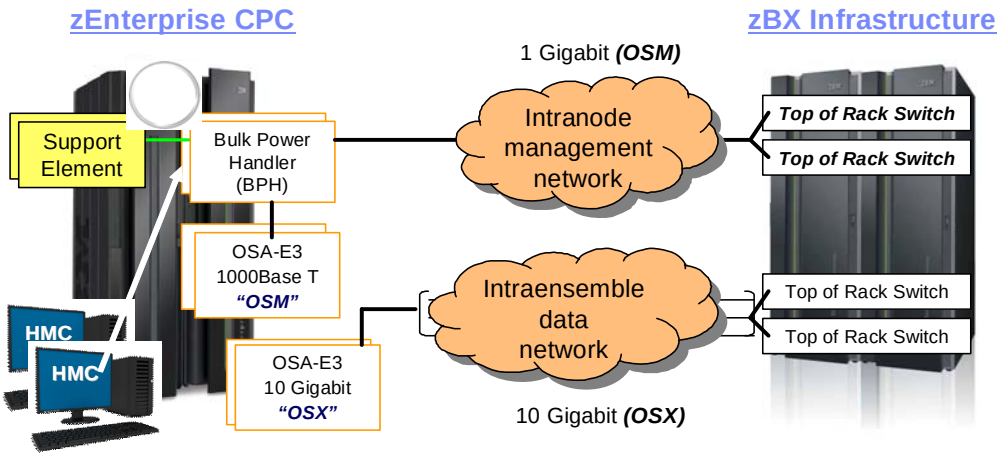
```
EZZ7847I ROUTING TABLE 796
TABLE NAME:     SECLOW2
TYPE   DEST NET         MASK      COST    AGE     NEXT HOP(S)

SBNT   8.0.0.0          FF000000  1       1549    NONE
 SPF   8.8.8.8          FFFFFFFC  2       1545    9.67.100.8
 SPF   8.8.8.8          FFFFFFFF  2       1545    9.67.100.8
SBNT   9.0.0.0          FF000000  1       1368    NONE
 DIR*  9.67.100.0       FFFFFF00  1       1576    9.67.100.7
 SPF   9.67.100.7       FFFFFFFF  2       1545    OSALINK2
 SPF   9.67.100.8       FFFFFFFF  1       1572    9.67.100.8
 SPF   9.67.105.4       FFFFFFFF  2       1545    9.67.100.8
SPE2   130.200.0.0      FFFF0000  0       1379    9.67.100.8 (2)
                      0 NETS DELETED

DYNAMIC ROUTING PARAMETERS:
  INTERFACE:  OSALINK2      NEXT HOP: 9.67.100.8
  INTERFACE:  OSALINK2      NEXT HOP: 9.67.100.15
  INTERFACE: *OSALINK3      NEXT HOP: 9.67.201.53
```


1. In this example, the PRtable=prname option has been used to display a particular policy-based route table. The prname value of SECLOW2 results in only that route table being displayed.
2. OMPROUTE has no knowledge of policy-based route tables that are defined without dynamic routing parameters. Those route tables are using static routing only. Since OMPROUTE has no knowledge of those tables, they cannot be displayed with the OMPROUTE DISPLAY command.
3. Table SECLOW2 is defined with three dynamic routing parameters that each specify a link and next hop. All dynamic routes added to this table should be either direct routes over one of these links or indirect routes over one of the links that have the associated IP address as next hop.
4. Most of the information in the display of a policy-based route table is the same as what is included in the display of the main route table. What is added for policy-based route tables is the name of the table at the top and the dynamic routing parameters being used for the table at the bottom.
5. The asterisk beside the link name in the last dynamic routing parameter shown in this example indicates that OSALINK3 is either not currently defined to the TCP/IP stack or not currently active. In either case, there would be no dynamic routes in the route table over that link.
6. When PRtable=ALL is specified, similar information is repeated for all of the OMPROUTE policy-based route tables.

Appendix: zEnterprise intraensemble data network (IEDN) & the OSX OSA Port

- Use the Network Virtualization Management function in the zEnterprise Unified resource Manager to enforce VLAN IDs and VMACs on the IEDN path.

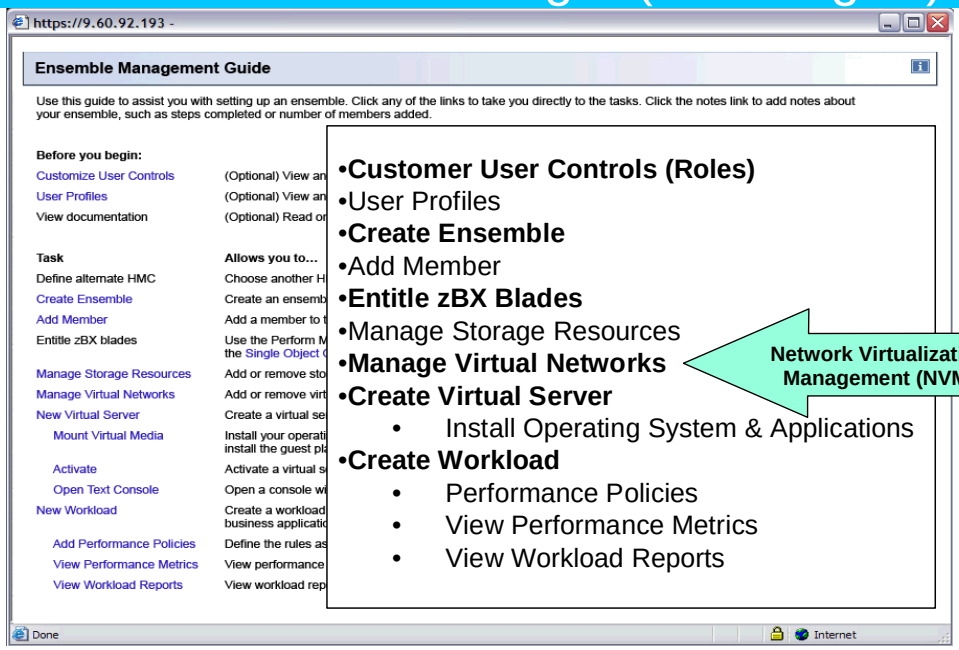
The zEnterprise™ System



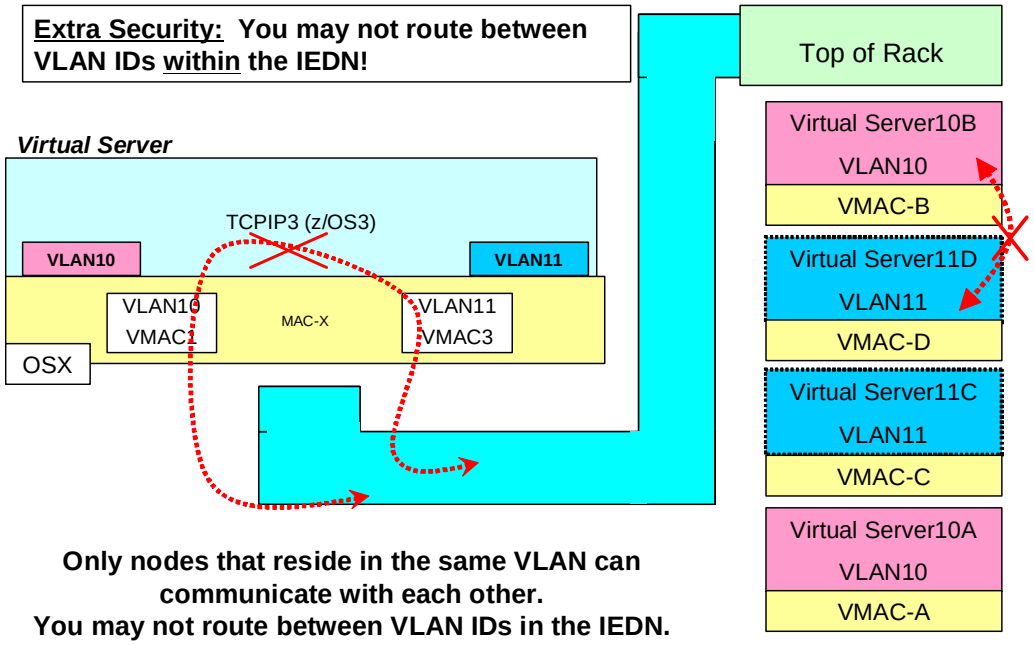
- Intranode management network (INMN)
  - 1000Base-T OSA-Express3 (copper) --- QDIO (***CHPID Type OSM***) – Cables are 3.2 meters long from OSM to BPH in CEC and 26 meters from BPH to TOR
  - HMC security is implemented with standard practices ***PLUS*** additional security mechanisms:
    - Isolated IPv6 network with ***"link-local"*** addresses only; authentication and authorization and access control, etc.
- Intraensemble data network (IEDN)
  - 10 Gigabit OSA-Express3 --- QDIO (***CHPID Type OSX***) – Cables are maximum of 26 meters long to TOR & 10km long-range
  - Security is implemented with standard practices ***PLUS*** additional security mechanisms: access control, authentication, authorization, application security, routing table restrictions, IP Filtering, etc.
  - Networks can be further isolated using VLAN and VMAC segmentation of the network connections

1. The zEnterprise System is comprised of: the z196 platform, the zBX, and the Unified Resource Manager (zManager).

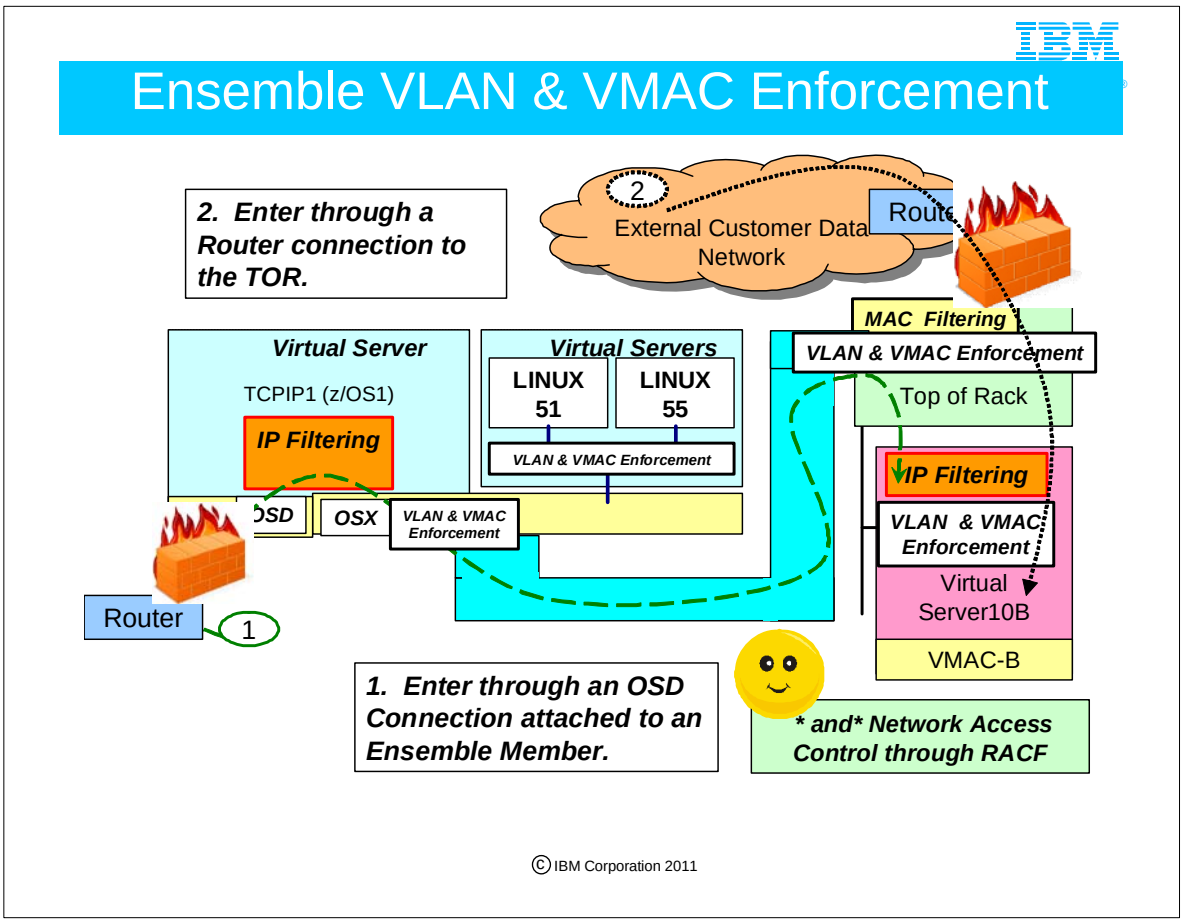
Creating an Ensemble with the HMC and Unified Resource Manager ("zManager")



Can You Route between VLAN IDs in an IEDN? No! VLAN Segmentation is Even More Secure



1. Normally Layer 3 IP Routing can be used to route between nodes on different VLANs. (Layer 2 routing between different VLANs is not possible and is not part of the 802.1q standard.) However, in the ensemble and on z/OS, the ability to invoke Layer 3 routiing over an OSX CHPID has been disabled to further strengthen the segmentation of traffic in the Ensemble. This design is important for the following reasons.
2. Over the years it has become widely accepted that VLANs are not viable forms of security, because both the MAC address and the VLAN ID can be spoofed or altered. However, in an IEDN these types of issues do not apply, because we control all access to the LAN via our hypervisors – including the OSX implementation as a VSwitch. (With OSX an Operating System cannot build its own Logical Link Control – “LLC” – header which includes the VLAN ID.)
3. Ensemble members that reside in a zBX may not be sharing ports at all. Even in such a situation, the VLAN ID prevents sending a message intended for one member to another. In this visual you see that the IEDN secure architecture does not permit routing through the z/OS stack in order to interconnect nodes on VLAN10 and VLAN11. Likewise, in the zBX, Virtual Server 10B on VLAN10 cannot communicate with Virtual Server11D on VLAN11.
4. Virtual Server Operating Systems should not forward from one VLAN ID to another on an OSX CHPID via Layer 3 routing. In fact, on z/OS IP Forwarding has been disabled when traffic comes in on an OSX port and wishes to exit on another OSX port using a different VLAN ID.
5. Both z/OS and z/VM can implement the ISOLATE function for an OSD or OSX interface; although ISOLATE is not necessary to isolate traffic over a Shared OSA implementation when one has deployed different VLAN IDs (as in the diagram above), you can still code ISOLATE on the Interface definition to prevent communication over a shared OSA port.
6. When anything is connected to the IEDN, a VLAN must be used. When the traffic hits the TOR port, the sending adapter (server) must have applied a VLAN ID tag. Therefore, for z/OS it will be tagged by the OSA at z/OS, and for an external router that wishes to connect to the TOR, the traffic must be tagged by that router.
7. VLAN ID enforcement adds a layer of security; it takes place at the Hypervisor.
8. hypervisor. A program that allows multiple instances of operating systems or virtual servers to run simultaneously on the same hardware device. A hypervisor can run directly on the hardware, can run within an operating system, or can be imbedded in platform firmware. Examples of hypervisors include PR/SM, z/VM, and PowerVM." In fact, even the OSX is considered a Hypervisor VSwitch, or a type of PR/SM Hypervisor. z/VM is a also a Hypervisor for a direct attachment to the OSX, where zVM itself is considered a type of PR/SM Hypervisor.
9. In the scenario depicted, the TOR IEDN Ports to the LPARs would be configured in TRUNK Mode because the Operating Systems are defining their VLAN IDs (are “VLAN-aware”). If z/VM is a host for a Virtual Server whose Operating System is not VLAN-aware, the z/VM VSWITCH – which is a Hypervisor – handles the VLAN ID on behalf of the Virtual Server. TOR ports to the Virtual Servers on the BLADEs are also configured in TRUNK mode. The TOR ports to any ISAOPT blades are configured in ACCESS Mode.
10. The hypervisor’s VSwitch -- whichever it may be (OSX for native z/OS or z/VM, VSWITCH for Virtual Servers on a VM VSWITCH, or the VSwitch on the blade, like pHype or xHype) -- performs the enforcement. The TOR performs the enforcement for anything that does not fall into these categories just mentioned.
11. The TOR performs the enforcement for anything that does not fall into these categories just mentioned.
12. For example, if a native LPAR is communicating with a virtual server on a blade, the OSX will perform VLAN enforcement, it passes through the TOR and the Ethernet Switch Module (ESM) without checking, but it is then checked again at the blade's VSwitch (=hypervisor).



1. If you decide to permit communication between the External Customer Data Network and the Ensemble, you can keep this path secure.
2. First, determine whether you want to do this.
3. Second, determine how you will secure this connection. Be aware of the fact that the TOR performs VLAN ID enforcement for connections to servers outside the zBX that are not attached to an OSX OSA port. The VLAN ID enforcement for any Virtual Server attached to an OSX works as follows: If a z/OS Native LPAR is on the OSX, then the OSX performs the VLAN ID enforcement; If the Virtual Server is under z/VM and attached to a VSWITCH, then the VSWITCH performs the VLAN ID enforcement. If an ISAOPT appliance is on the zBX, then the TOR performs the VLAN ID enforcement. The hypervisors on the Blade of the Virtual Servers of the zBX perform VLAN ID Enforcement in the zBX.
4. You configure the authorized VLAN IDs at the HMC as part of the Network Virtualization infrastructure.
5. Remember that Security protection is much more than just inserting a firewall along a path. It encompasses all layers of the IP Stack: Application Security Mechanisms (Access Control Lists, Userid and Password checking, mapping mechanisms), Transport Security Mechanisms (SSL/TLS, AT-TLS), IP Layer Security Mechanisms (IPSec, IP Filtering, Intrusion Detection Services, Network Address Tables), Data Link Control Security Mechanisms (MAC Address Filtering, VLAN Segmentation or Segregation), and many more mechanisms too numerous to mention here.
6. With regard to MAC Filtering, the HMC performs MAC filtering for external MAC Addresses. The MACs within the IEDN are managed by the Network Virtualization Manager. All MACs are allowed that originate from within the IEDN (they are managed by NVM). z/VM VSwitch MAC Protect function for Layer 2 is on by default for IEDN type VSwitches. This MAC Protect function enforces that a VMAC sent during guest link initialization (SETVMAC) matches with what has been assigned by the zVM hypervisor. In addition, all SOURCE MAC addresses on egress frames from the guest are verified to insure that only the assigned VMAC for the guest is being sent on outbound data transfers. This eliminates any attempt by the guest to spoof its source MAC address.
7. If you are still concerned about firewalls consider these possibilities:
    1. Firewalls
        1. IP Filtering in z/OS Policy Agent (not stateful)
        2. IP Filtering with Proventia Intrusion Prevention Services for Linux on z (not stateful)
        3. IP Filtering in Virtual Servers residing on the zBX (may or may not be stateful)
        4. Other IP Filtering mechanisms (could be stateful or not)
        5. Firewall in front of LPAR that is attached to an OSD OSA
        6. Firewall in front of TOR in External network
        7. SERVAUTH Classes: NETACCESS CONTROLS for IEDN
        8. MAC ADDRESS FILTERING at the TOR
        9. MultiLevel Security

Appendix
Advanced Topics

VLAN Terminology

- Tagged Frame: A frame tagged with an 802.1q VLAN header
- Untagged Frame: A frame not tagged with an 802.1q VLAN header
- Default VLAN: A default VLAN is a VLAN assigned to an access port for stacks that are VLAN unaware. The default VLAN ID is often set to "1." The stacks send and receive untagged traffic from the switch's access port. If a stack on a trunk port (VLAN aware) sends a frame to a stack on an access port (VLAN unaware) the switch will strip the tag before delivering it to the stack on the access port. When a stack on an access port sends a frame to the stack on a trunk port the switch tags the frame with the default VLAN ID. When a stack on an access port sends a frame to the stack on a trunk port the switch will tag the frame with the default VLAN ID and deliver it to the trunk port.
- Native VLAN: Native VLANs are used by switches to flow untagged traffic. The frames remain untagged but are delivered to trunk ports that are configured for the native vlan or untagged traffic
- Null VLAN: nulls VLANs are used for priority queuing (user priority queueing with 802.1p)

Background Notes

- How are VLANs (802.1q) used in a shared-OSA environment and how routing with VLANs occurs over a shared OSA. Some general concepts about VLANs are:
  - VLANs, or Virtual LANs, are a Layer 2 method of segmenting networks by assigning a VLAN Identifier (VLAN ID) to subnets of the network. In this way, what formerly looked like a shared medium (the network LAN segment), now looks like separate LAN segments, even though all the segments may terminate in the same shared adapter. In this paper, that shared adapter is the OSA port of the z platform.
  - Therefore, separate VLANs should be configured in separate subnets.
  - Layer 3 routing protocols (static, dynamic) route according to IP addresses and their subnets. Neither a routing table nor an IP stack takes a VLAN ID into consideration when building or acting upon entries in the routing tables. Therefore, the OSPF routing protocol does not know about VLANs and VLAN IDs.
  - Network switches do make decisions based upon the Layer 2 concept of the VLAN ID.
  - An OSA port -- after having examined the MAC -- primarily delivers an inbound packet to a destination based upon its destination IP address or the PRIRouter and SECRouter definitions in an IP stack. You cannot code PRI- or SECRouter if you have coded the VMAC parameter. (If the interface is coded with VMAC, the IP Address is ignored unless ROUTELCL is used.) If a VLAN ID has been received, then the VLAN ID may or may not be matched against an IP address depending on whether a VMAC is present or not. However, once the OSA has selected a target based on the two criteria mentioned, it then subjects the delivery of the packet to an examination of a VLAN ID, if one has been provided.
- Due to the factors just presented, special considerations must be made when designing a shared OSA environment on a z platform with mixed operating systems.
- This differing behavior between Linux on z and the other two operating systems (z/OS and z/VM) has consequences for a network design using shared OSAs and VLAN IDs.

Comparing VLANs in z/OS, z/VM and Linux on z

- z/OS and z/VM may establish VLAN IDs in their interface definitions. However, whether or not they have established a VLAN ID on an OSA definition, they are still considered "VLAN-unaware" stacks.
  - If a z/OS or z/VM stack has not implemented a VLAN definition on an OSA port, then the stack is obviously VLAN-unaware.
  - If a z/OS or z/VM stack has implemented a VLAN definition on an OSA port, the stack remains VLAN-unaware, although the terms "VLAN-unaware" and "VLAN-aware" are ambiguous to many and tend to be misused when applied to z/OS and z?VM.
    - This is due to a VLAN architecture called "Global VLAN."
    - When the OSA port is activated on the stack, the VLAN ID is registered in that port. **From that moment on, the stack ignores VLAN IDs and allows the OSA to manage the delivery of packets based upon VLAN ID.**
      - With a "Global VLAN" any IP frames are stripped of their VLAN tags prior to being delivered to the operating system by the OSA port.
- Linux on z is either "VLAN-unaware" or "VLAN-aware."
  - If Linux is VLAN-unaware, no definition for a VLAN ID has been configured for the connection to the OSA port.
    - As a result, Linux is completely unaware of a VLAN ID and does not want to see one on inbound packets.
  - If Linux is VLAN-aware, a definition of a VLAN ID has been configured for the OSA device.
    - When the OSA port is activated on the Linux stack, the VLAN ID is registered in that port. **However, the Linux remains aware of the VLAN ID. Linux accepts or discards inbound frames based upon the presence or absence of a VLAN tag in the frame header.**
      - The OSA preserves VLAN tags before delivering a packet to the Linux stack.

Global VLAN Stacks vs. VLAN-aware Stacks without VMAC Routeall Coding **

- Global VLAN is a concept for z/OS and z/VM
  - OSA provides the filtering for packets inbound to z/OS or z/VM based on the VLAN ID.
    - Untagged frame: If OSA finds that the inbound frame is untagged, then it attempts to match the destination IP address. If a match is found, the IP packet is delivered. If a matching IP address is not found, then the packet is delivered to the stack designated as PriRouter or SecRouter. If there is neither a matching IP address nor a PriRouter or SecRouter designation, then the packet is dropped.
    - Tagged frame: If OSA finds a matching destination IP address, it verifies that the VLAN tag is appropriate for the matching stack; it strips the VLAN tag and sends to the z/OS or z/VM stack with the Global VLAN ID. If a matching IP address is not found, it delivers the packet to the PRIRouter or SECRouter image associated with the appropriate VLAN ID after stripping the VLAN tag. If there is neither a matching IP address nor a PRIRouter or SECRouter designation associated with the VLAN ID of the tag, then the packet is dropped.
  - **Conclusion: A tagged or an untagged packet can be accepted by a stack coded with Global VLAN (z/OS or z/VM). z/OS and z/VM don't care about VLANIDs for inbound packets, but the OSA does.**
- The OSA behaves in a different fashion with LINUX on z, because Linux does not use Global VLAN.
  - It does not strip the tag before handing it to Linux.
  - If an untagged packet arrives inbound to Linux, and Linux is VLAN-aware, then the packet is discarded.
  - If a tagged packet arrives inbound to Linux, and Linux is not VLAN-aware, then the packet is discarded.
  - **Conclusion: Linux on z cares about VLAN IDs of inbound packets. If VLAN-aware, it can receive only packets that are correctly VLAN-tagged; if not VLAN-aware, it cannot receive tagged packets.**

** Please see Table of OAT registrations in this presentation for information on the effect of VMAC Routeall vs. VMAC Routelcl.

End of Topic

End of Topic