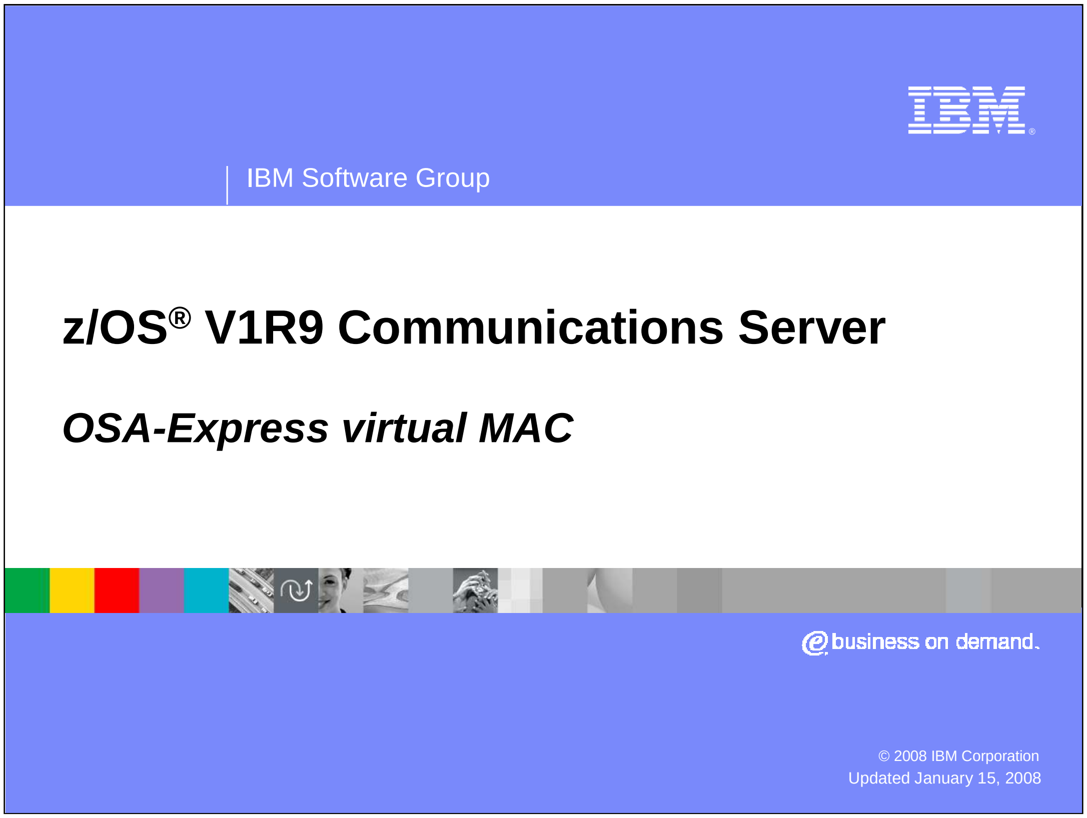


IBM Software Group

# z/OS® V1R9 Communications Server

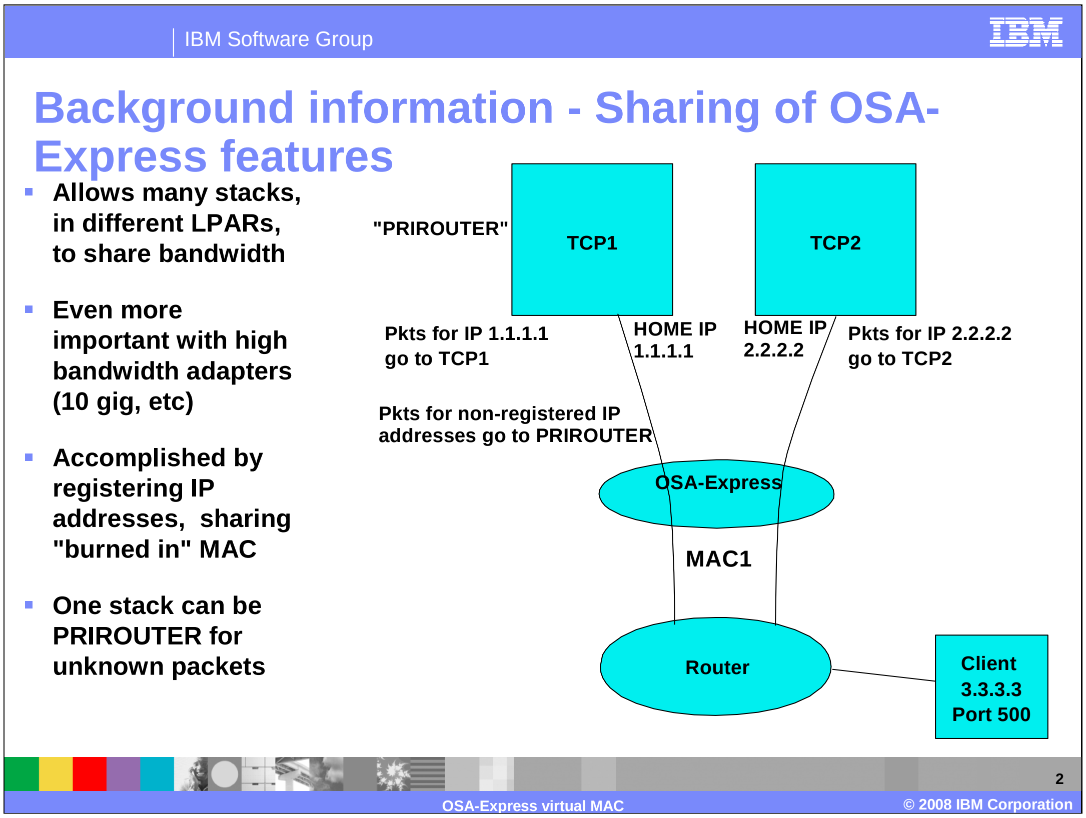
## *OSA-Express virtual MAC*

business on demand.

© 2008 IBM Corporation

Updated January 15, 2008

This presentation discusses the OSA-Express virtual MAC function in the z/OS V1R9 Communications Server.



IBM Software Group

## Background information - Sharing of OSA-Express features

- Allows many stacks, in different LPARs, to share bandwidth
- Even more important with high bandwidth adapters (10 gig, etc)
- Accomplished by registering IP addresses, sharing "burned in" MAC
- One stack can be PRIROUTER for unknown packets

"PRIROUTER"

TCP1

TCP2

Pkts for IP 1.1.1.1 go to TCP1

HOME IP 1.1.1.1

HOME IP 2.2.2.2

Pkts for IP 2.2.2.2 go to TCP2

Pkts for non-registered IP addresses go to PRIROUTER

OSA-Express

MAC1

Router

Client 3.3.3.3 Port 500

2

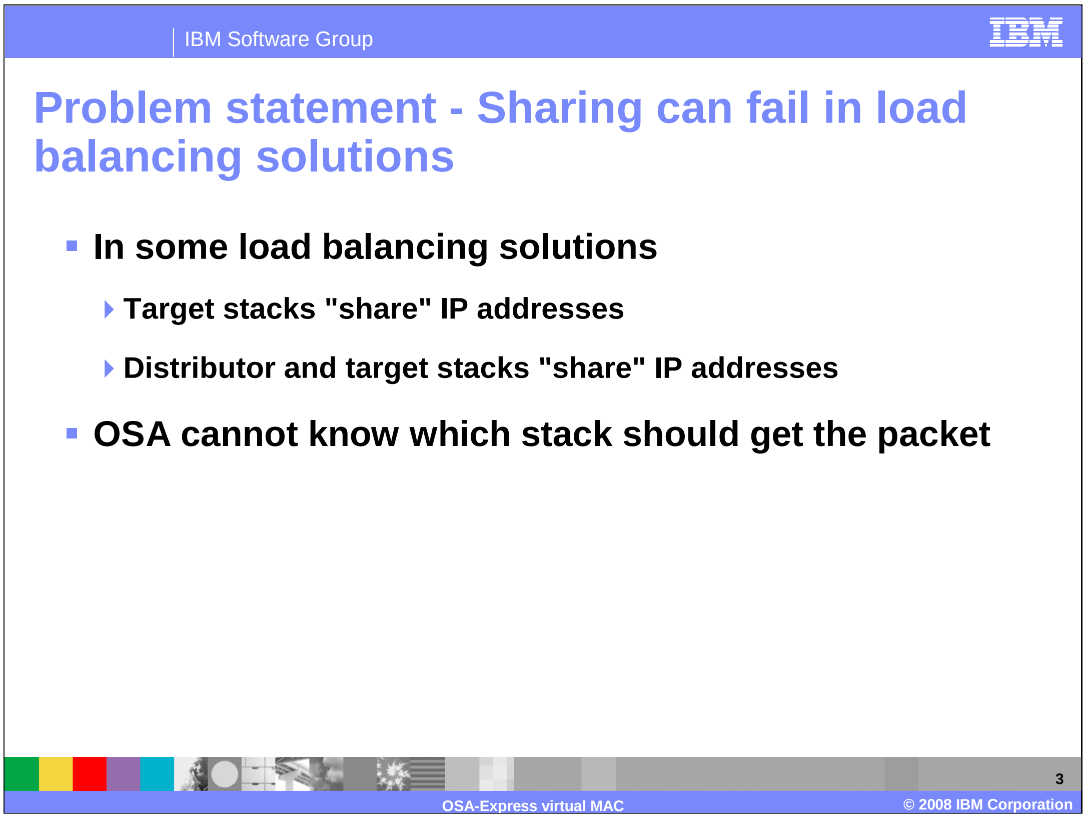
OSA-Express virtual MAC

© 2008 IBM Corporation

With high bandwidth adapters, one stack on one LPAR typically does not send or receive enough traffic to fully use all the bandwidth of the OSA. To get the money for your investment, you want multiple stacks on multiple LPARs using, or sharing, the same OSA.

The figure on the right is an example of how sharing works today. In this example, stack 1 registers its home IP addresses for this OSA, such as 1.1.1.1, while stack 2 registers its home IP addresses, including 2.2.2.2. The OSA ARPs all these addresses using its one physical burned in MAC, so everyone on the LAN knows to get to any of these addresses, use that MAC. Then the OSA routes to the correct stack using the IP address. It knows that everything to 1.1.1.1 goes to TCP1, and everything to 2.2.2.2 goes to TCP2.

The stack that is PRIROUTER will get any packet sent to an IP address that is not registered in the OSA.

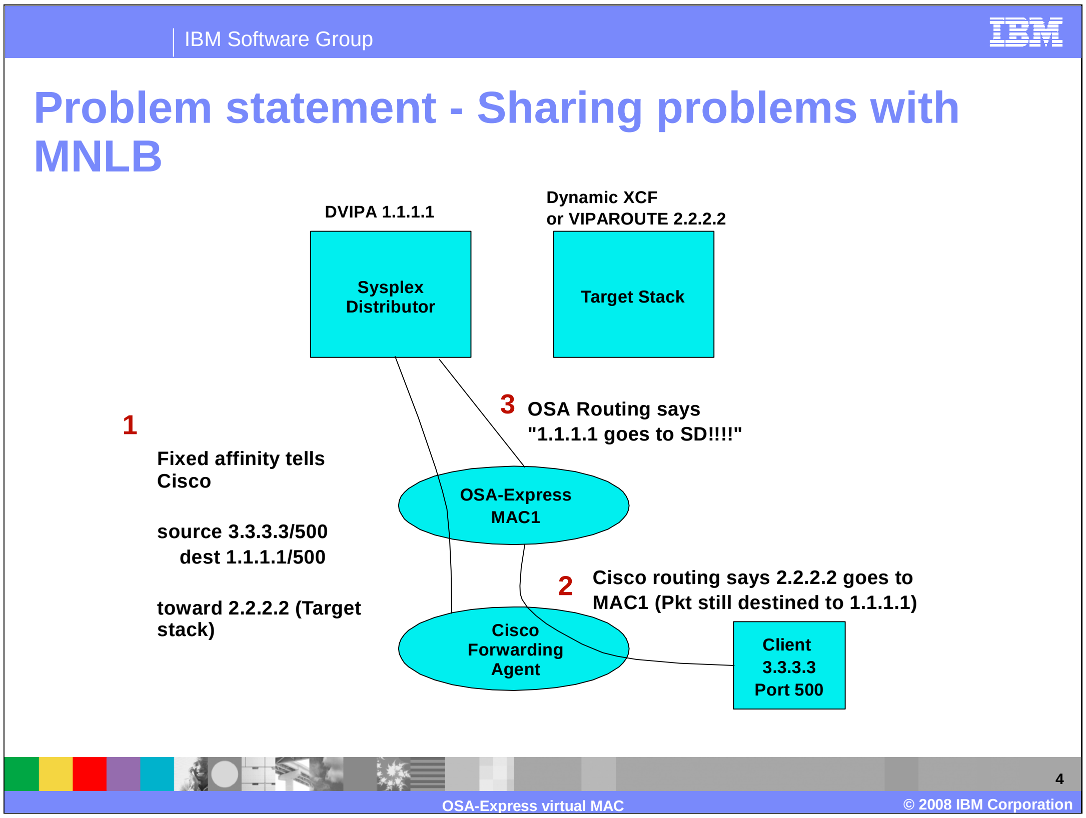


## Problem statement - Sharing can fail in load balancing solutions

- In some load balancing solutions
  - Target stacks "share" IP addresses
  - Distributor and target stacks "share" IP addresses
- OSA cannot know which stack should get the packet

© 2008 IBM Corporation

The next few charts will show some examples of load balancing solutions that share IP addresses, and how load balancing fails in these environments.



IBM Software Group

IBM

## Problem statement - Sharing problems with MNLB

DVIPA 1.1.1.1

Sysplex Distributor

Dynamic XCF
or VIPAROUTE 2.2.2.2

Target Stack

1

Fixed affinity tells Cisco

source 3.3.3.3/500
dest 1.1.1.1/500

toward 2.2.2.2 (Target stack)

3 OSA Routing says "1.1.1.1 goes to SD!!!!"

OSA-Express MAC1

2 Cisco routing says 2.2.2.2 goes to MAC1 (Pkt still destined to 1.1.1.1)

Cisco Forwarding Agent

Client
3.3.3.3
Port 500

4

OSA-Express virtual MAC

© 2008 IBM Corporation

MNLB is Multinode Load Balancer. In Multinode Load Balancing, the Sysplex Distributor registers load balancing information with the  routers acting as forwarding agents. In particular, it registers a 4-tuple connection to the Cisco router, and which target IP address the Cisco should use to find the target stack MAC to send all data for that connection.

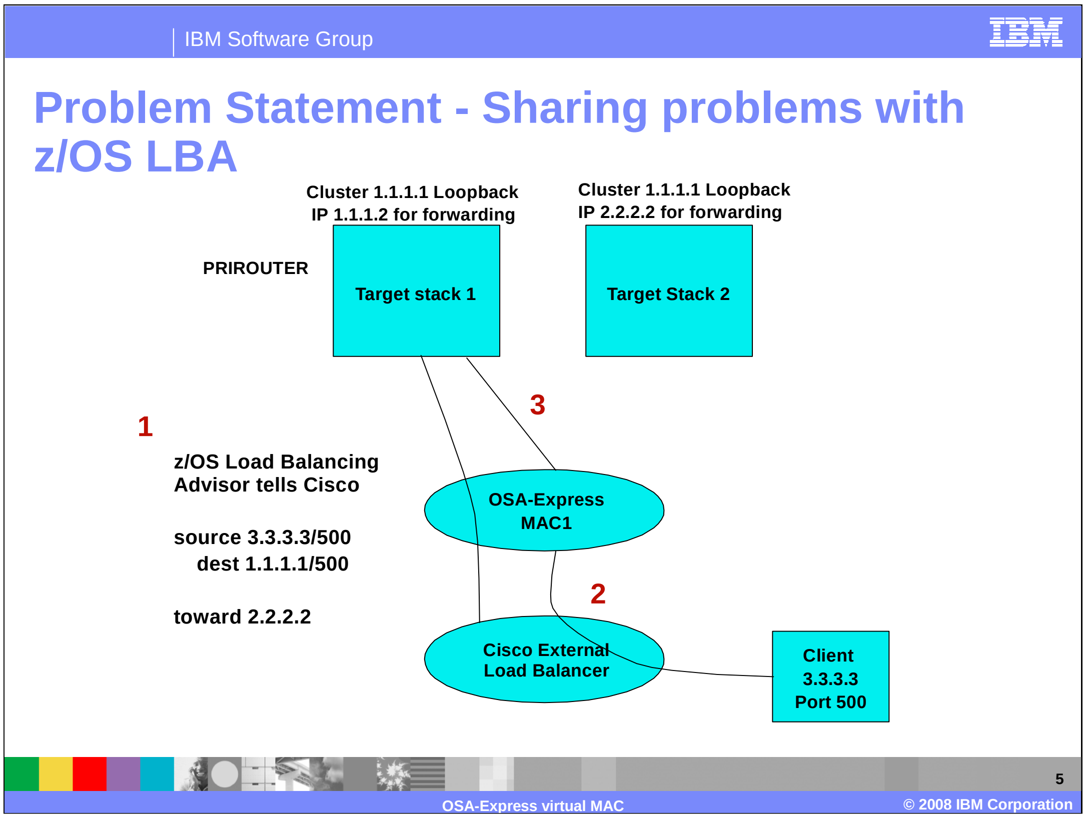
The problem is that the Cisco forwarding agents use the same MAC address for packets destined to the Distributor stack or the Target stack. So the OSA has know way to know that some packets destined to the DVIPA should go to the distributor because there is no current affinity to a target stack, and some should go to the target stack because an affinity is already established.

In an MNLB configuration, TCP/IP's Sysplex Distributor informs the Cisco Forwarding Agents that a given 4-tuple should be sent "toward" a particular target stack IP address - either the dynamic XCF address or the VIPAROUTE address of the target stack. The Sysplex Distributor does this by knowing which target stack it assigned to this connection. In this example, the Cisco knows to send any packet from 3.3.3.3, port 500, to DVIPA 1.1.1.1, port 500, toward destination IP address 2.2.2.2.

Cisco, when it sees a packet for this 4-tuple, uses its routing table to know how to get to that target stack IP address. It then forwards that packet toward the MAC of that target stack IP address, but does not change the contents of this packet. In this example, the Cisco would see target destination IP address 2.2.2.2 can be reached through MAC1. So it would send the packet to MAC1, but the packet still has the original 4-tuple, and in particular, the DVIPA1 address as the destination IP address.

Since OSA gets the packet with the original 4-tuple, it sees the destination IP address as DVIPA1. Since the Distributor stack has registered DVIPA1, the OSA will forward the packet to the distributor, even though the Cisco intended the packet to go directly to the target stack.

The conclusion is that because the 2 target stacks share the same MAC, with MNLB there is no way for the OSA to know which packets are truly destined to the distributor and which should go to the target stack.



IBM Software Group

## Problem Statement - Sharing problems with z/OS LBA

Cluster 1.1.1.1 Loopback
IP 1.1.1.2 for forwarding

Cluster 1.1.1.1 Loopback
IP 2.2.2.2 for forwarding

PRIROUTER

Target stack 1

Target Stack 2

3

1

z/OS Load Balancing
Advisor tells Cisco

source 3.3.3.3/500
dest 1.1.1.1/500

toward 2.2.2.2

OSA-Express
MAC1

2

Cisco External
Load Balancer

Client
3.3.3.3
Port 500

5

OSA-Express virtual MAC

© 2008 IBM Corporation

z/OS LBA is Communication Server’s load balancing advisor. In LBA, the Load Balancing Advisor registers the load balancing information with the  routers acting as external load balancers, such as a Cisco Content Switching Module (CSM).  In particular, in dispatch mode, the Load Balancing Advisor registers a 4-tuple connection to the Cisco external load balancer, and which target IP address the Cisco should use to find the target stack MAC to send all data for that connection.

The problem is that the Cisco external load balancer configured in dispatch mode uses the same MAC address for packets destined to the two target stacks. So the OSA has know way to know that some packets destined to the cluster IP address 1.1.1.1 already have an affinity to target stack 1 and should go there, and some have an affinity with target stack to and should go there.

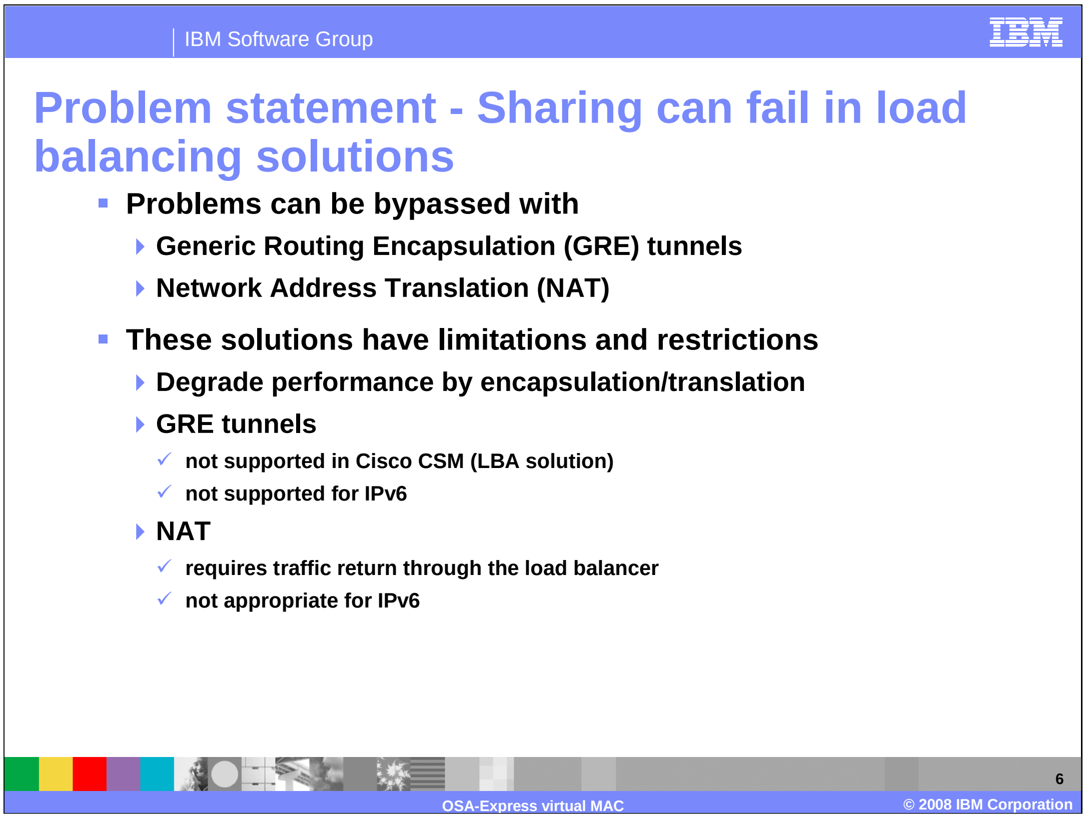
z/OS Load Balancing Advisor is subject to the same problems with shared OSAs as MNLB. This is particularly true when the external load balancer is in dispatch mode. In dispatch mode, as with MNLB, the external load balancer does not alter the packet, but forwards the packet to the MAC of a destination IP address. In a shared OSA configuration, the following problem can occur.

TCP/IP's z/OS Load Balancing Advisor informs the Cisco external load balancer that a given 4-tuple should be sent "toward" a particular target stack IP address. That IP address is the IP address of a target stack it learned from the z/OS Load Balancing agent.  In this example, the Cisco knows to send any packet from 3.3.3.3, port 500, to cluster IP address 1.1.1.1, port 500, toward destination IP address 2.2.2.2

Cisco, when it sees a packet for this 4-tuple, uses its routing table to know how to get to that target stack IP address. It then forwards that packet toward the MAC of that target stack IP address, but does not change the contents of this packet. In this example, the Cisco would see target destination IP address 2.2.2.2 can be reached through MAC1. So it would send the packet to MAC1, but the packet still has the original 4-tuple, and in particular, the cluster IP address as the destination IP address.

Since OSA gets the packet with the original 4-tuple, it sees the destination IP address as a cluster IP address that has not been registered. The OSA will therefore forward the packet to the stack defined as the PRIROUTER,  even though the Cisco intended the packet to go directly to target stack 2.

The conclusion is that because the 2 target stacks share the same MAC, with LBA in dispatch mode there is no way for the OSA to know to which target stack the packet is truly destined. For this reason, any customers sharing OSAs and running LBA typically use directed mode. This has the overhead of doing Network Address Translation.



IBM Software Group

## Problem statement - Sharing can fail in load balancing solutions

- Problems can be bypassed with
  - Generic Routing Encapsulation (GRE) tunnels
  - Network Address Translation (NAT)
- These solutions have limitations and restrictions
  - Degrade performance by encapsulation/translation
  - GRE tunnels
    - ✓ not supported in Cisco CSM (LBA solution)
    - ✓ not supported for IPv6
  - NAT
    - ✓ requires traffic return through the load balancer
    - ✓ not appropriate for IPv6

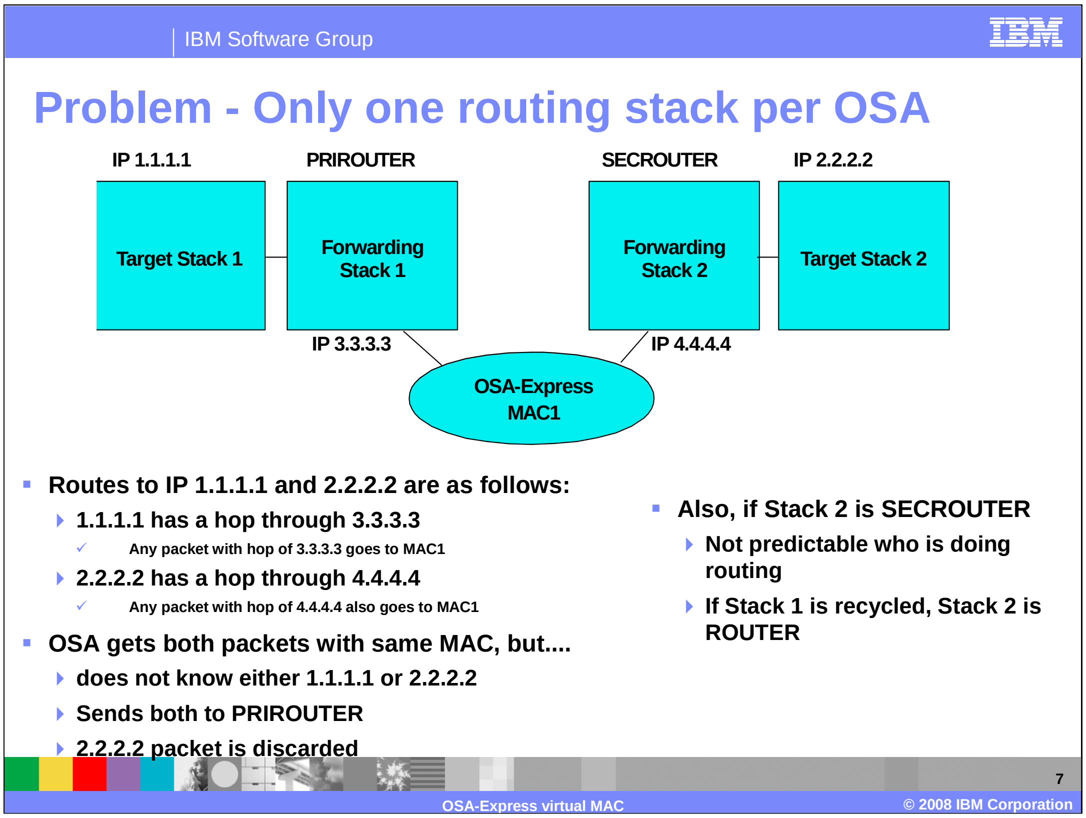
6

OSA-Express virtual MAC

© 2008 IBM Corporation

The previous problems can be solved by using either GRE tunnels or using load balancing advisors configured for directed mode.  For MNLB, GRE tunnels are configured from the Cisco Forwarding Agents to the target stacks.  Thus, Cisco will imbed the packets for a given 4-tuple in another packet (GRE encapsulated packet) with the destination IP address of the target stack.  For z/OS LBA, either OSAs are not shared, or the external load balancer is configured in directed mode. In directed mode, the destination IP addresses are converted using Network Address Translation (NAT) to IP addresses that belong to the given target stack.  The bypass of using GRE tunnels is not supported in Cisco CSM, and thus cannot be used for LBA. Even if an external load balancer did support GRE tunnels for IPv4, GRE tunnels are not architected in IPv6.

Network Address Translation (NAT) can be used for LBA by defining the external load balancer in directed mode. This allows the external load balancer to actually change the destination IP address from the cluster IP address to an IP address on the correct target stack. However, NAT requires that the return traffic go through the external load balancer, which is not always the best route for the return traffic, and burdens the router acting as the external load balancer. With NAT, there is also an overhead associated with changing the IP address of every packet. And one of the advantages of going to IPv6 is that NAT is not required, like it sometimes is when IPv4 addresses are exhausted in a given customer shop. When LBA and IPv6 are used together, however, this again requires NAT with shared OSAs.



IBM Software Group

# Problem - Only one routing stack per OSA

IP 1.1.1.1

PRIROUTER

SECROUTER

IP 2.2.2.2

Target Stack 1

Forwarding Stack 1

Forwarding Stack 2

Target Stack 2

IP 3.3.3.3

IP 4.4.4.4

OSA-Express MAC1

- Routes to IP 1.1.1.1 and 2.2.2.2 are as follows:
  - 1.1.1.1 has a hop through 3.3.3.3
    - ✓ Any packet with hop of 3.3.3.3 goes to MAC1
  - 2.2.2.2 has a hop through 4.4.4.4
    - ✓ Any packet with hop of 4.4.4.4 also goes to MAC1
- OSA gets both packets with same MAC, but....
  - does not know either 1.1.1.1 or 2.2.2.2
  - Sends both to PRIROUTER
  - 2.2.2.2 packet is discarded

- Also, if Stack 2 is SECROUTER
  - Not predictable who is doing routing
  - If Stack 1 is recycled, Stack 2 is ROUTER

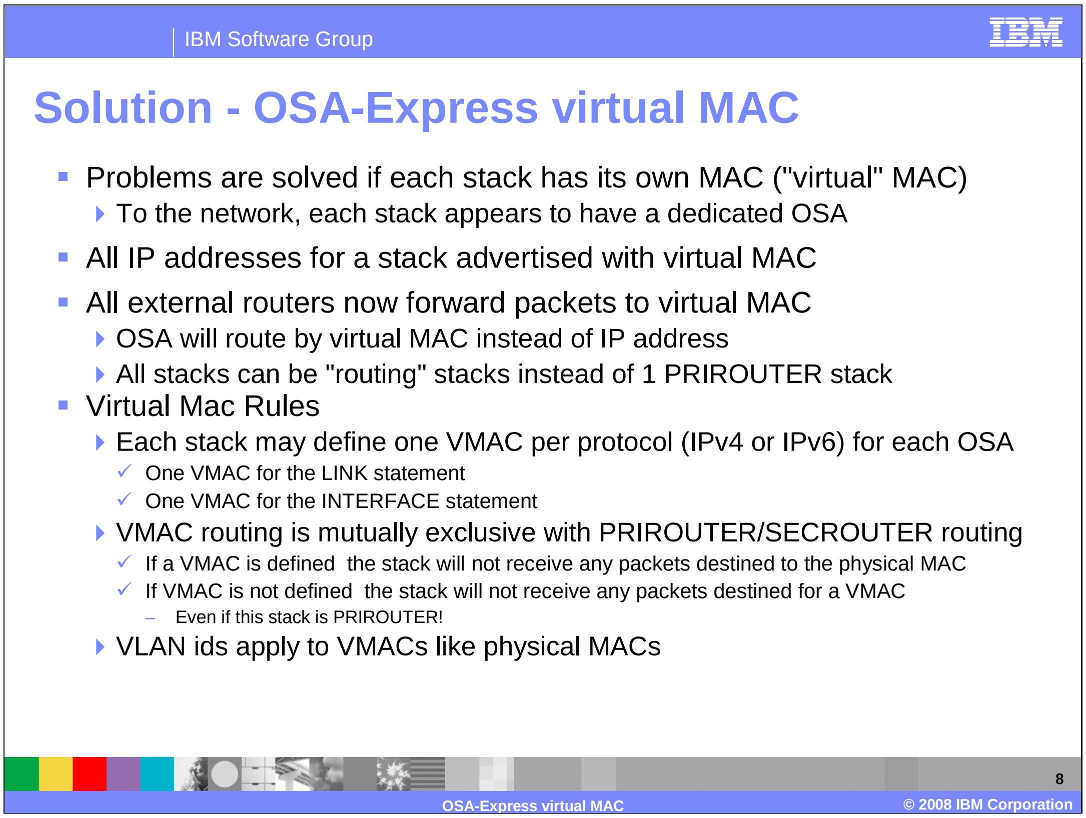
7

OSA-Express virtual MAC

© 2008 IBM Corporation

Though not related to load balancing, there is another problem with OSAs shared by multiple stacks. This is if  two stacks sharing an OSA both act as routing stacks.

Note that in this example the OSA does not have either the address of target stack 1 or the address of target stack 2 registered, because neither stack is directly connected to it. Therefore, the OSA forwards packets destined to either target stack to the PRIROUTER stack called forwarding stack 1. Because of this, packets destined to target stack 2 will never reach target stack 2.

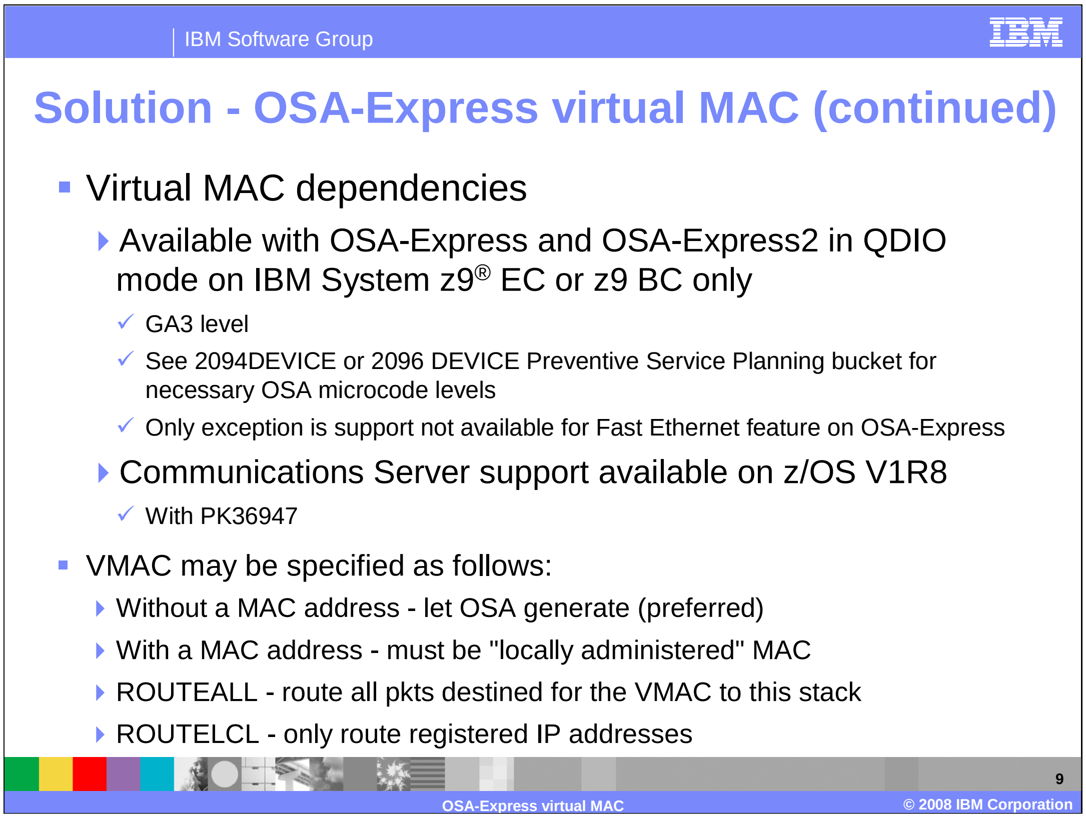


## Solution - OSA-Express virtual MAC

- Problems are solved if each stack has its own MAC ("virtual" MAC)
  - To the network, each stack appears to have a dedicated OSA
- All IP addresses for a stack advertised with virtual MAC
- All external routers now forward packets to virtual MAC
  - OSA will route by virtual MAC instead of IP address
  - All stacks can be "routing" stacks instead of 1 PRIROUTER stack
- Virtual Mac Rules
  - Each stack may define one VMAC per protocol (IPv4 or IPv6) for each OSA
    - ✓ One VMAC for the LINK statement
    - ✓ One VMAC for the INTERFACE statement
  - VMAC routing is mutually exclusive with PRIROUTER/SECROUTER routing
    - ✓ If a VMAC is defined  the stack will not receive any packets destined to the physical MAC
    - ✓ If VMAC is not defined  the stack will not receive any packets destined for a VMAC
      - Even if this stack is PRIROUTER!
  - VLAN ids apply to VMACs like physical MACs

All these problems are resolved if each stack has its own MAC address.   All IP addresses for a stack are advertised with its virtual MAC by OSA using ARP for IPv4 and by the stack using Neighbor Discovery (ND) for IPv6.

The  VMACs for IPv4 are defined on the LINK statement representing the OSA-Express. The VMACs for IPv6 are defined on the INTERFACE statement representing the OSA-Express.  You can specify a VMAC on the LINK statement for IPv4, and use the same VMAC or a different VMAC on the INTERFACE statement for IPv6. You can also specify a VMAC on one statement (LINK or INTERFACE), and not on the other, thus using a VMAC for one protocol and the physical MAC for the other. Also, one stack can use a VMAC for its connection to the OSA, and another stack can use the physical MAC.  VMACs may be assigned to a particular VLAN id, just like the physical MAC could.



# Solution - OSA-Express virtual MAC (continued)

- Virtual MAC dependencies
  - Available with OSA-Express and OSA-Express2 in QDIO mode on IBM System z9® EC or z9 BC only
    - ✓ GA3 level
    - ✓ See 2094DEVICE or 2096 DEVICE Preventive Service Planning bucket for necessary OSA microcode levels
    - ✓ Only exception is support not available for Fast Ethernet feature on OSA-Express
  - Communications Server support available on z/OS V1R8
    - ✓ With PK36947
- VMAC may be specified as follows:
  - Without a MAC address - let OSA generate (preferred)
  - With a MAC address - must be "locally administered" MAC
  - ROUTEALL - route all pkts destined for the VMAC to this stack
  - ROUTELCL - only route registered IP addresses

© 2008 IBM Corporation

This virtual MAC function is only available on a z9 at the GA3 level, but both the OSA-Express and OSA-Express2 platforms have the virtual MAC function available. Note that the function and the publications were available in z/OS V1R8, but until GA3 of the z9 EC and BC, the OSA code was not available. The virtual MAC function is available with Communications Server V1R8 when APAR PK36947 is applied.

When VMAC is specified without a MAC address, the OSA will generate one. OSA's VMAC generation scheme, to guarantee uniqueness, is as follows:

> First byte of VMAC will be a constant 02. The 2 bit indicates this is a locally administered MAC address. This will guarantee it is unique from all physical "burned-in" MACs, since the 2 bit for these adapters is off, indicating they are "universal" addresses. The last 3 bytes will be the last 3 bytes of the physical MAC address. This will guarantee all VMACs on one OSA will be unique from all other VMACs on any other OSA. To guarantee stacks sharing an OSA will get unique addresses, the second and third bytes of the VMAC will be an instance count, incremented each time OSA gives out a VMAC address.

OSA will generate a different VMAC for IPv6 versus IPv4. The same generation rules apply as applied to the LINK statement. The VMAC generated for the INTERFACE statement should differ from that generated for the DEVICE/LINK statement only by the instance count.

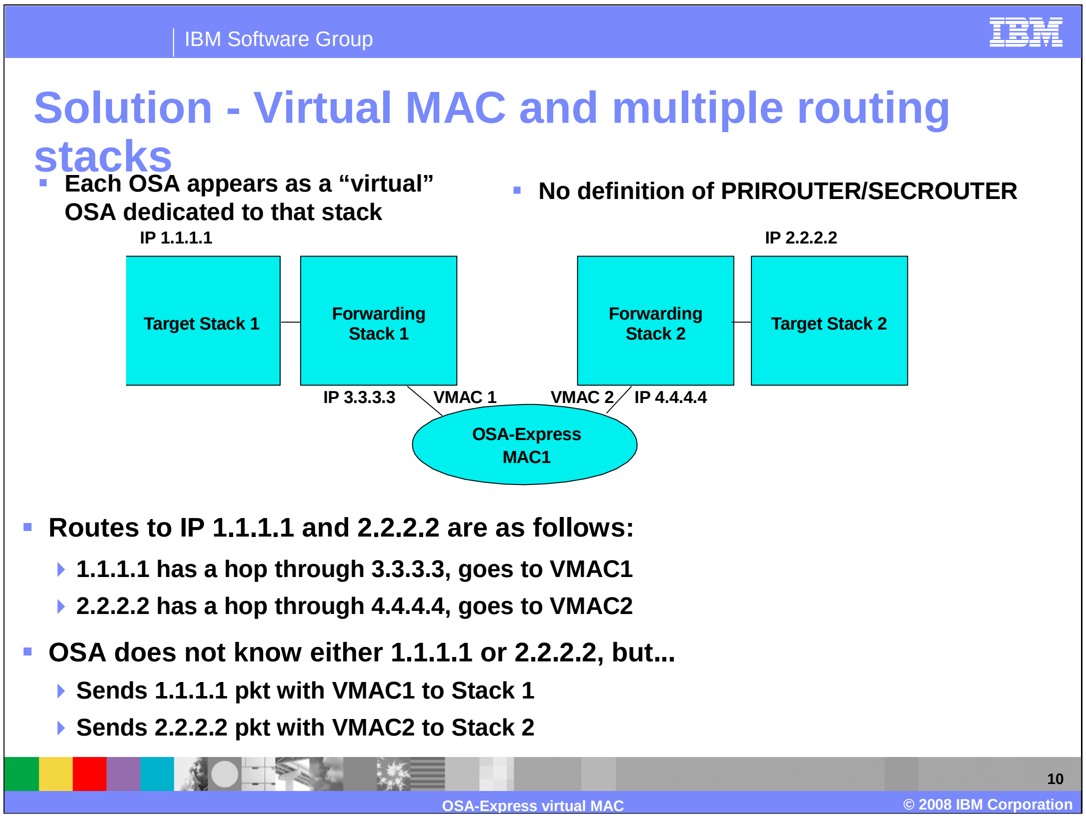
If INTFID is not defined by the user, on the IPv6 INTERFACE statement, the generated INTFID is different for VMAC versus non-VMAC INTERFACES. For non-VMAC INTERFACES, it is first 3 bytes of MAC, followed by OSA generated instance count, followed by last 3 bytes of MAC. This is because the MAC is shared by multiple INTERFACEs, and the INTFID needs to be unique. For VMAC INTERFACEs, the VMAC is unique for each stack. So the standard form of an interface ID is generated. This is the first 3 bytes of the VMAC, followed by X'FFFE', followed by the last 3 bytes of the VMAC.

TCP/IP will reuse the same generated VMAC address when a device becomes inactive and is reactivated. A new VMAC address will be generated for a given OSA if the stack is stopped and restarted.

If the VMAC is defined by the user, it must be a 12 digit hexadecimal number, with the **X'02'** bit in the first byte of the VMAC on, indicating this is a locally administered MAC address. It is up to the user to ensure the uniqueness of the VMAC on the local LAN on which this OSA resides. It is recommended VMACs be used anytime the OSA is shared.

When ROUTEALL is specified or defaulted on the LINK or INTERFACE statement, then all packets destined for the VMAC are routed to this stack. This is done even if the IP address is not registered. When ROUTELCL is specified on the LINK or INTERFACE statement that only packets for registered IP addresses will be routed to this stack. This parameter should only be used when the stack will not forward OSA traffic.

The following charts will show how the virtual MAC function solves all 3 major issues. Multiple routing stacks, MNLB, and LBA.



IBM Software Group

## Solution - Virtual MAC and multiple routing stacks

- Each OSA appears as a “virtual” OSA dedicated to that stack
- No definition of PRIROUTER/SECROUTER

IP 1.1.1.1
IP 2.2.2.2
Target Stack 1
Forwarding Stack 1
Forwarding Stack 2
Target Stack 2
IP 3.3.3.3
VMAC 1
VMAC 2
IP 4.4.4.4
OSA-Express MAC1

- Routes to IP 1.1.1.1 and 2.2.2.2 are as follows:
  - 1.1.1.1 has a hop through 3.3.3.3, goes to VMAC1
  - 2.2.2.2 has a hop through 4.4.4.4, goes to VMAC2
- OSA does not know either 1.1.1.1 or 2.2.2.2, but...
  - Sends 1.1.1.1 pkt with VMAC1 to Stack 1
  - Sends 2.2.2.2 pkt with VMAC2 to Stack 2

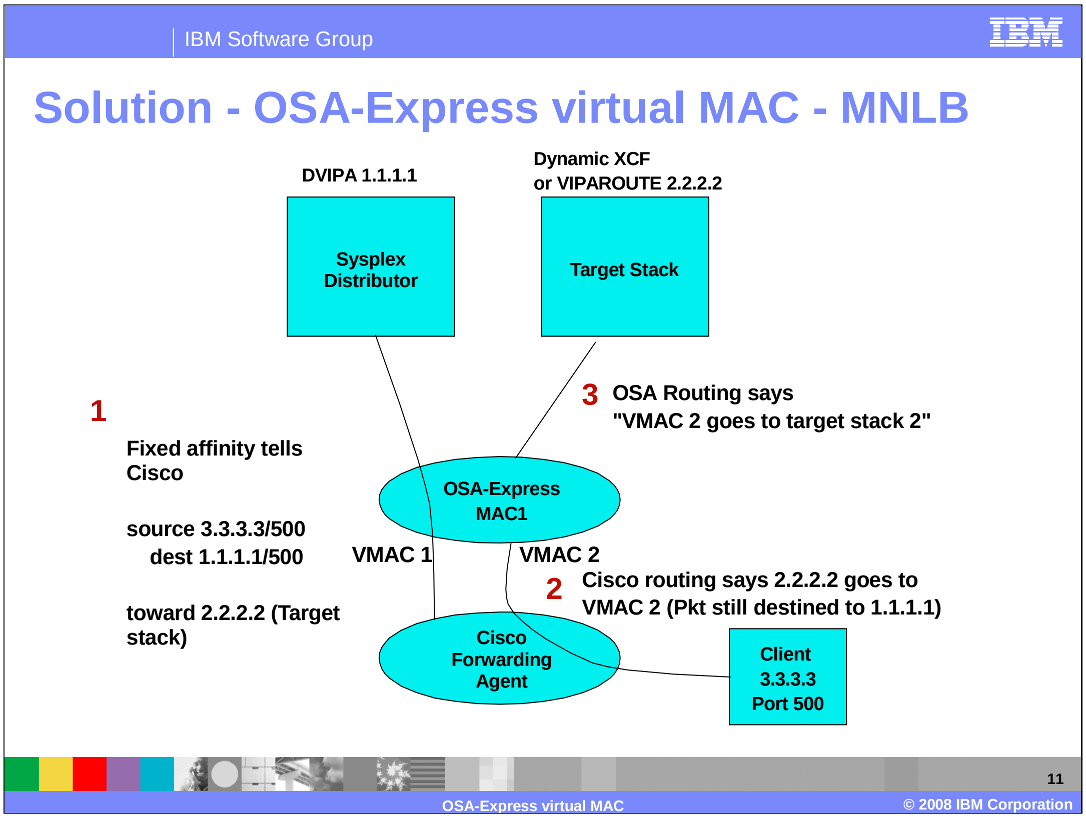
10

OSA-Express virtual MAC

© 2008 IBM Corporation

Remember the multiple routing stack problem? Virtual MACs solve this problem.

Since now the packet destined for 1.1.1.1 has a route through 3.3.3.3, its destination MAC address when the OSA gets it will not be the burned in MAC, but will be VMAC1. Therefore, the OSA knows to forward that packet to Forwarding stack 1. Likewise, the packet destined for 2.2.2.2 has a destination MAC of VMAC2, so the OSA knows to forward that packet to Forwarding stack 2.



IBM Software Group

## Solution - OSA-Express virtual MAC - MNLB

DVIPA 1.1.1.1

Sysplex Distributor

Dynamic XCF
or VIPAROUTE 2.2.2.2

Target Stack

1

Fixed affinity tells Cisco

source 3.3.3.3/500
dest 1.1.1.1/500

toward 2.2.2.2 (Target stack)

3 OSA Routing says "VMAC 2 goes to target stack 2"

OSA-Express
MAC1

VMAC 1

VMAC 2

2 Cisco routing says 2.2.2.2 goes to VMAC 2 (Pkt still destined to 1.1.1.1)

Cisco Forwarding Agent

Client
3.3.3.3
Port 500

11

OSA-Express virtual MAC

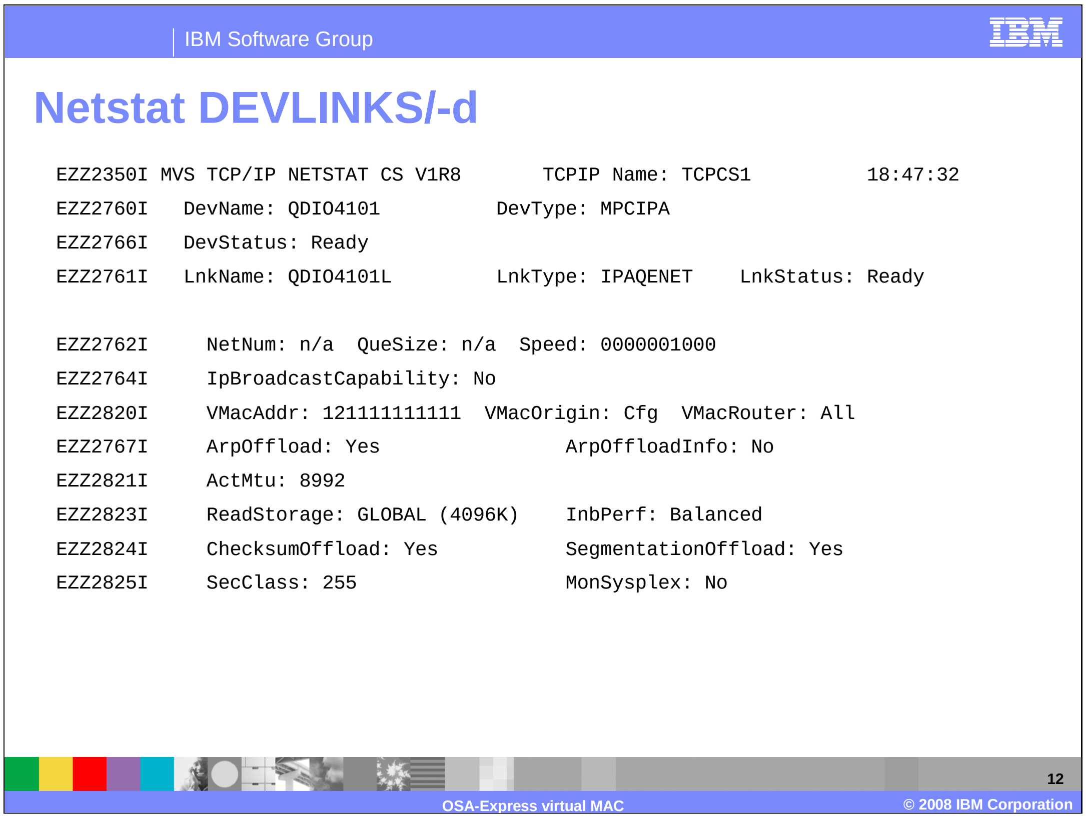
© 2008 IBM Corporation

Let’s see how VMAC fixes the MNLB problems. In an MNLB configuration with VMACs, TCP/IP's Sysplex Distributor informs the Cisco Forwarding Agents that a given 4-tuple should be sent "toward" a particular target stack IP address - either the dynamic XCF address or the VIPAROUTE address of the target stack. The Sysplex Distributor does this by knowing which target stack it assigned to this connection. In this example, the Cisco knows to send any packet from 3.3.3.3, port 500, to DVIPA 1.1.1.1, port 500, toward destination IP address 2.2.2.2.

Cisco, when it sees a packet for this 4-tuple, uses its routing table to know how to get to that target stack IP address. It then forwards that packet toward the MAC of that target stack IP address. However, with VMAC in place, the OSA has advertised that destination IP address 2.2.2.2 can be reached not through shared MAC1, but through unique VMAC2. So it would send the packet to VMAC2, still with the original 4-tuple, and in particular, the DVIPA1 address as the destination IP address.

OSA gets the packet with the original 4-tuple, but the destination is not shared MAC1, but VMAC2. Because the packet is to a VMAC, it will route the packet directly to the target stack owning that VMAC, even though the destination IP address of DVIPA1 is registered to the distributor stack.

VMAC provides these same advantages for z/OS Load Balancing Advisor solutions, and in some configurations allows for using dispatch mode instead of directed mode for the external load balancer. Dispatch mode will still be subject to some special considerations when the load balancer is more than one hop away from the target systems. See the IP Configuration Guide, section "External IP workload balancing solutions," for more details on these considerations.



IBM Software Group

# Netstat DEVLINKS/-d

```
EZZ2350I MVS TCP/IP NETSTAT CS V1R8       TCPIP Name: TCPCS1         18:47:32
EZZ2760I   DevName: QDIO4101          DevType: MPCIPA
EZZ2766I   DevStatus: Ready
EZZ2761I   LnkName: QDIO4101L         LnkType: IPAQENET    LnkStatus: Ready

EZZ2762I     NetNum: n/a  QueSize: n/a  Speed: 0000001000
EZZ2764I     IpBroadcastCapability: No
EZZ2820I     VMacAddr: 121111111111  VMacOrigin: Cfg  VMacRouter: All
EZZ2767I     ArpOffload: Yes               ArpOffloadInfo: No
EZZ2821I     ActMtu: 8992
EZZ2823I     ReadStorage: GLOBAL (4096K)    InbPerf: Balanced
EZZ2824I     ChecksumOffload: Yes          SegmentationOffload: Yes
EZZ2825I     SecClass: 255                 MonSysplex: No
```

12

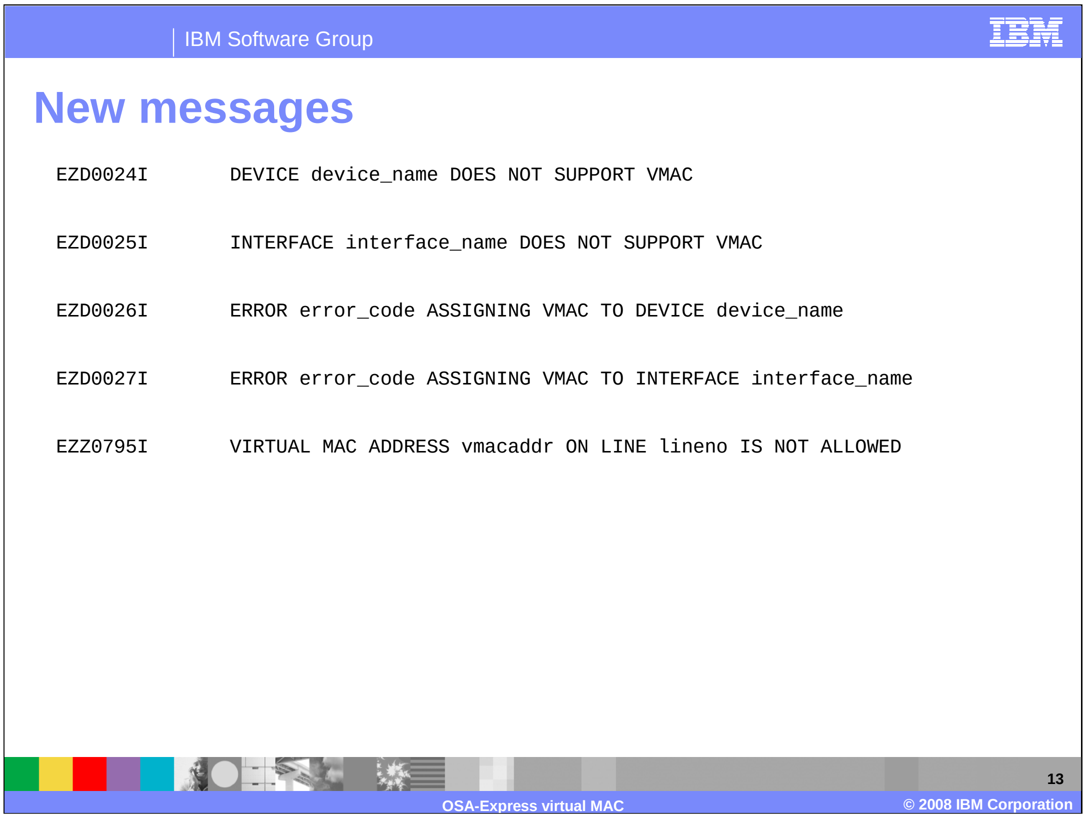
OSA-Express virtual MAC

© 2008 IBM Corporation

The VMAC fields are displayed on the netstat devlinks command.

The VMacAddr field will be either the predefined VMAC address if VMAC xxxxxxxx was defined, or the OSA generated VMAC address if VMAC was defined without a specific MAC address.

The VMacOrigin field will be Cfg if the MAC address is defined by the user, or OSA if the MAC address is generated by the OSA.
The VMacRouter field will be ALL if ROUTEALL was specified or defaulted for the VMAC, or LCL if ROUTELCL was specified for this VMAC.



## New messages

| | |
|---|---|
| EZD0024I | DEVICE device_name DOES NOT SUPPORT VMAC |
| EZD0025I | INTERFACE interface_name DOES NOT SUPPORT VMAC |
| EZD0026I | ERROR error_code ASSIGNING VMAC TO DEVICE device_name |
| EZD0027I | ERROR error_code ASSIGNING VMAC TO INTERFACE interface_name |
| EZZ0795I | VIRTUAL MAC ADDRESS vmacaddr ON LINE lineno IS NOT ALLOWED |

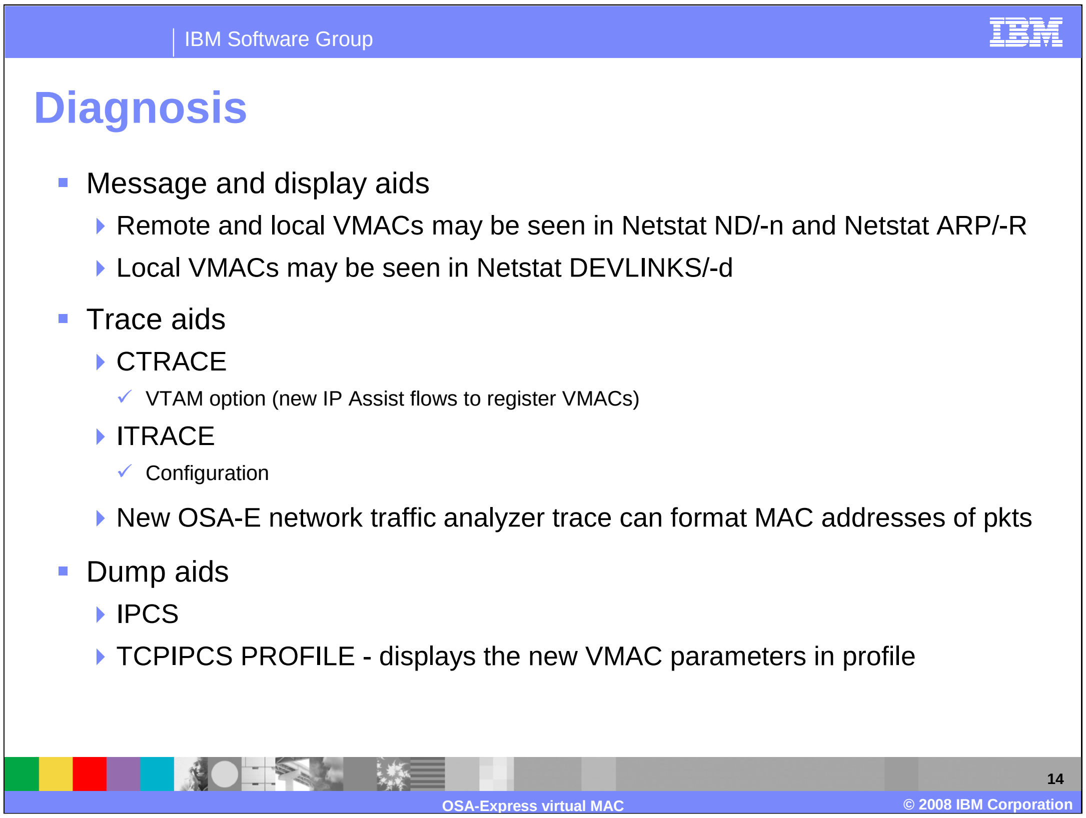
© 2008 IBM Corporation

These are the new messages introduced with this function.

Note that if VMAC is defined, and either the OSA does not support VMAC, or an error was reported attempting to assign the VMAC, Device or Interface activation fails. This is because it is assumed if VMAC was configured, other configurations were altered to use it. Those altered configurations will likely fail without VMAC. For example, the external load balancer may have been reconfigured to use dispatch mode, or GRE tunnels may have been removed from Cisco forwarding agents for MNLB. In either case, load balancing will now fail.

Messages EZD0026I and EZD0027I are expected to be issued with only one code - the code indicating the VMAC attempting to be assigned was already defined.

Message EZZ0795I is issued when the user attempts to configure a virtual MAC address with the local/universal bit set off, meaning universal MAC address. Only locally administered MAC addresses should be defined for VMAC. Device or interface configuration processing fails, for the same reasons stated with messages EZD0024I-EZD0027I.



IBM Software Group

## Diagnosis

- Message and display aids
  - Remote and local VMACs may be seen in Netstat ND/-n and Netstat ARP/-R
  - Local VMACs may be seen in Netstat DEVLINKS/-d
- Trace aids
  - CTRACE
    - ✓ VTAM option (new IP Assist flows to register VMACs)
  - ITRACE
    - ✓ Configuration
  - New OSA-E network traffic analyzer trace can format MAC addresses of pkts
- Dump aids
  - IPCS
  - TCPIPCS PROFILE - displays the new VMAC parameters in profile

14

OSA-Express virtual MAC

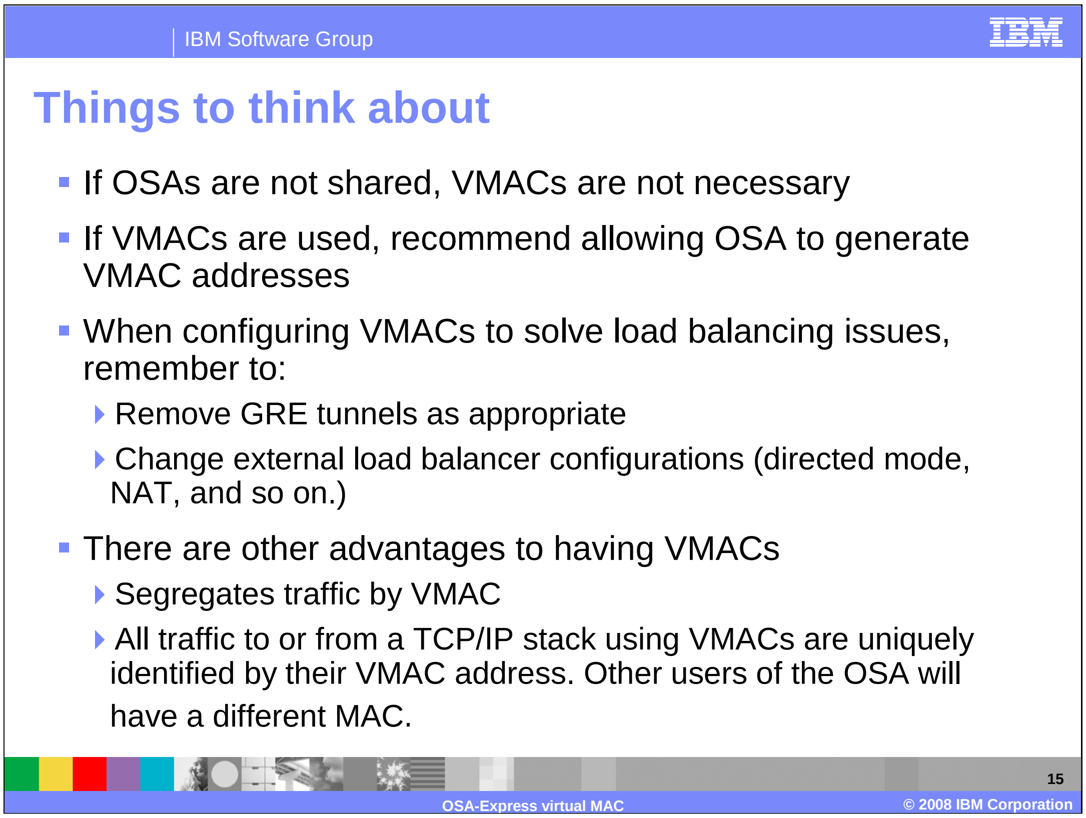
© 2008 IBM Corporation

Remote MAC addresses appear in several places. In the netstat arp and neighbor discovery displays, the remote MAC address is displayed with the remote IP address that is associated with. If the remote IP address is for an OSA that has been defined with a VMAC address, that MAC address will be a VMAC address.

The local MAC address appears on a netstat devlinks display. The local MAC address will likewise be the VMAC address if VMAC is defined for this OSA for this stack.

The new OSA reject codes can be located in the OSA-Express Customer's Guide and Reference.

Existing trace options in CTRACE and ITRACE should be sufficient to debug problems with registration of VMAC addresses. The new OSA Express Network Traffic Analyzer trace can be used to filter and format packets using these VMAC addresses, including seeing the ARP packets using these VMAC addresses.



IBM Software Group

# Things to think about

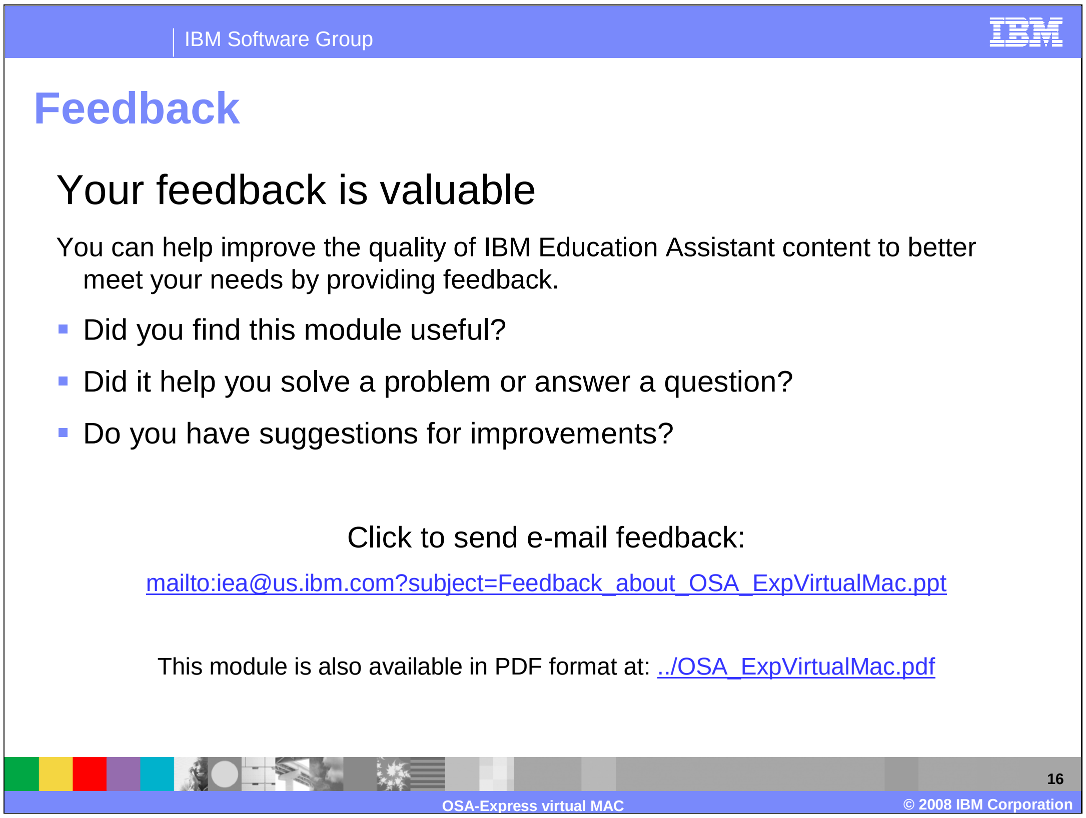
- If OSAs are not shared, VMACs are not necessary
- If VMACs are used, recommend allowing OSA to generate VMAC addresses
- When configuring VMACs to solve load balancing issues, remember to:
  - Remove GRE tunnels as appropriate
  - Change external load balancer configurations (directed mode, NAT, and so on.)
- There are other advantages to having VMACs
  - Segregates traffic by VMAC
  - All traffic to or from a TCP/IP stack using VMACs are uniquely identified by their VMAC address. Other users of the OSA will have a different MAC.

15

OSA-Express virtual MAC

© 2008 IBM Corporation

If OSAs are shared, VMACs should be used.



# Feedback

## Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

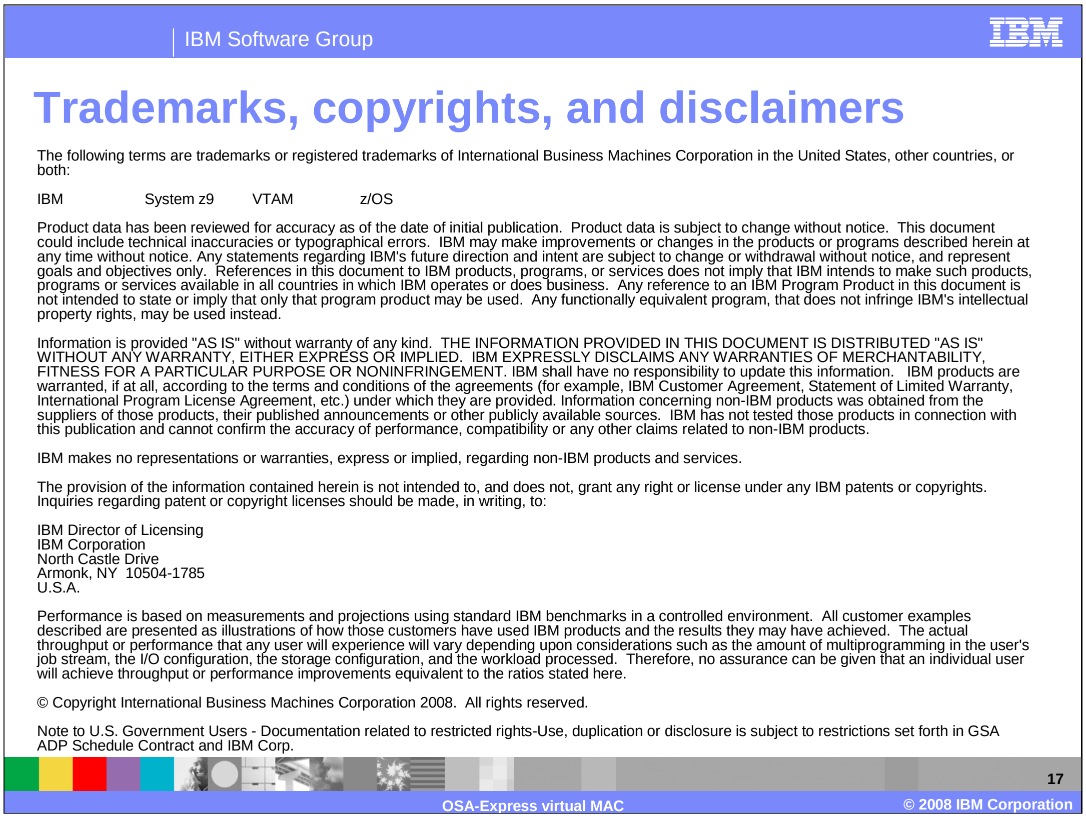
Click to send e-mail feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_OSA_ExpVirtualMac.ppt

This module is also available in PDF format at: ../OSA_ExpVirtualMac.pdf

© 2008 IBM Corporation

You can help improve the quality of IBM Education Assistant content by providing feedback.



# Trademarks, copyrights, and disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM System z9 VTAM z/OS

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2008. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.