Smarter Systems for a Smarter Planet

SHARE Atlanta 2012

Session 10413

Avoiding Common IPv6 Implementation Mistakes

Speaker: ***Gwendolyn J. Dente (gdente@us.ibm.com)***
IBM Advanced Technical Support (ATS)
Gaithersburg, Maryland (USA)

Friday, March 16, 2012
8:00 AM – 9:00 AM

(Chestnut Room at the OMNI CNN)

IBM Advanced Technical Skills (ATS) 1

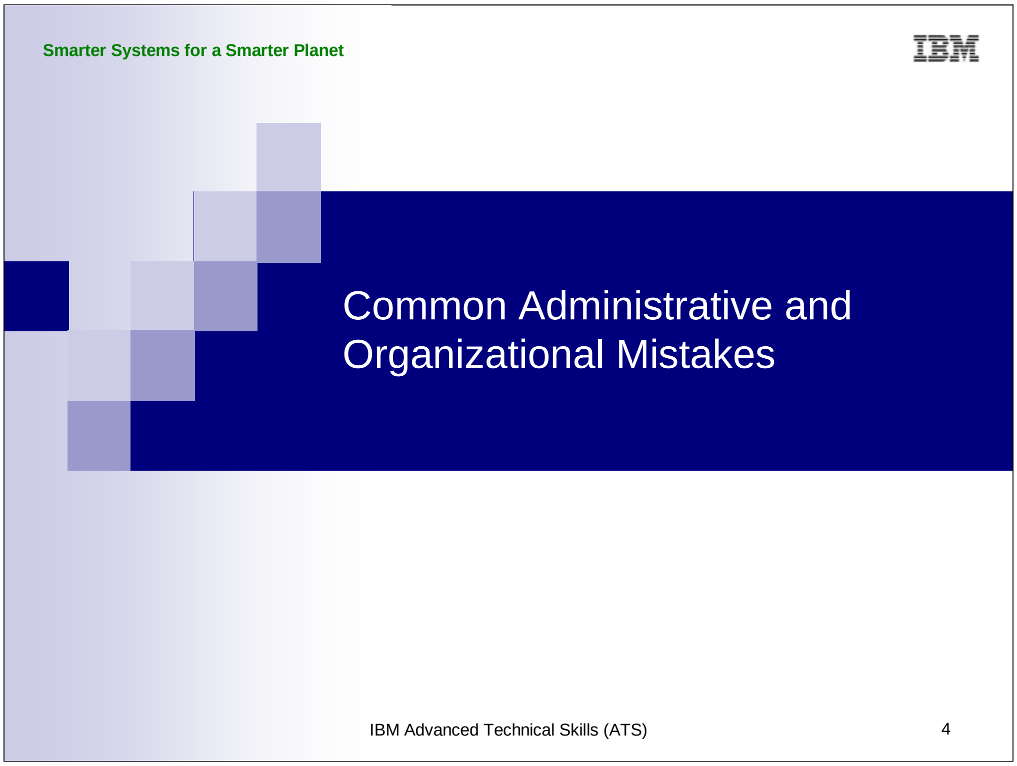
IBM

Abstract

- You have been provided with an LPAR to create an IPv6 network in preparation for the inevitable changes from IPv4 to IPv6. You are now attempting to design this network to use as a model for what you could eventually move into production. What common mistakes should you be aware of so that you can avoid them in designing and implementing your IPv6 network? This session represents both an implementation task plan and a list of the errors you want to avoid making.

Agenda

- **Common Administrative and Organizational Mistakes: Failing to …**
  - Read the simple documentation and planning presentations (web, Share, etc)
  - Gain executive commitment and buy-in to IPv6 testing
  - Acquire a test bed for IPv6 (hardware and software)
  - Create a cross-platform team to discuss testing environment
- **Common Technical Mistakes: Failing to …**
  - Enable IPv6 in z/OS prior to Implementing IPv6
  - Exploit HiperSockets and OSA Enhancements from last releases
    - Convert existing IPv4 QDIO Device/Link coding to INTERFACE coding
    - Exploit HiperSockets zIIP and Multiwrite enhancements
  - Convert Gateway routing statements to Beginroutes/Endroutes routing statements
  - Implement Global Resolver with a Resolver Setup File
  - Implement an IPNodes file for testing and optionally a DNS to handle IPv6
  - Design IPv6 addressing scheme for testing that looks to the future
    - False start: Attempting to use IPv4 numbering in IPv6 addresses
    - False start: “inventing” IPv6 addresses instead of exploiting existing addressing tools
    - If possible, integrating stateless auto-configuration and DHCP
  - Understand Source IP Address selection for local clients
- **Sample IPv6 Implementation Plan for Test Environment**
  - Including working with cross-platform team to determine the type of addressing prefix that provides the most flexibility for your permanent, non-test deployment of IPv6 addresses
    - Prefix assigned on a geographic basis
    - Prefix assigned by company


Smarter Systems for a Smarter Planet
IBM
Common Administrative and Organizational Mistakes
IBM Advanced Technical Skills (ATS)
4

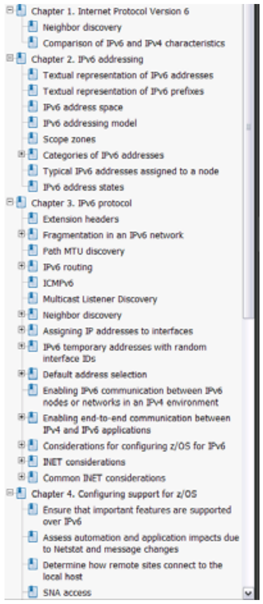
IBM

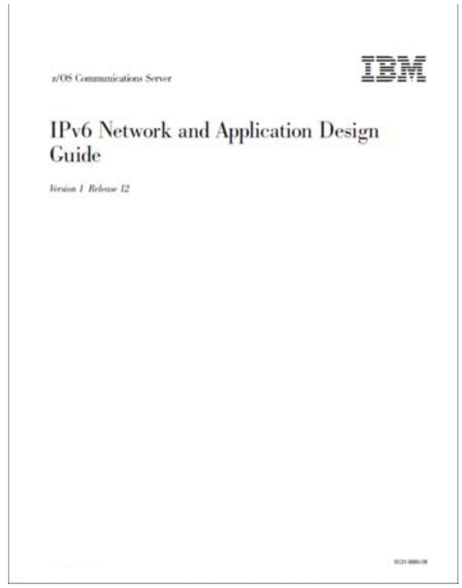
Agenda

- **Common Administrative and Organizational Mistakes: Failing to …**
  - Read the simple documentation and planning presentations (web, Share, etc)
  - Gain executive commitment and buy-in to IPv6 testing
  - Acquire a test bed for IPv6 (hardware and software)
  - Create a cross-platform team to discuss testing environment
- **Common Technical Mistakes: Failing to …**
  - Enable IPv6 in z/OS prior to Implementing IPv6
  - Exploit HiperSockets and OSA Enhancements from last releases
    - Convert existing IPv4 QDIO Device/Link coding to INTERFACE coding
    - Exploit HiperSockets zIIP and Multiwrite enhancements
  - Convert Gateway routing statements to Beginroutes/Endroutes routing statements
  - Understand Source IP Address selection for local clients
  - Implement Global Resolver with a Resolver Setup File
  - Implement an IPNodes file for testing and optionally a DNS to handle IPv6
  - Design IPv6 addressing scheme for testing that looks to the future
    - False start: Attempting to use IPv4 numbering in IPv6 addresses
    - False start: "inventing" IPv6 addresses instead of exploiting existing addressing tools
    - If possible, integrating stateless autoconfiguration and DHCP
- **Sample IPv6 Implementation Plan for Test Environment**
  - Including working with cross-platform team to determine the type of addressing prefix that provides the most flexibility for your permanent, non-test deployment of IPv6 addresses
    - Prefix assigned on a geographic basis
    - Prefix assigned by company

IBM Advanced Technical Skills (ATS) 5

Learn How to Design & Implement IPv6 Networks





IBM Advanced Technical Skills (ATS) 6

IPv6 Sessions at Share in Atlanta, March 2012

10395: IPv6 Tunneling Technologies	Wednesday, March 14th: 1:30pm
10397: IPv6 Basics	Wednesday, March 14th: 8:00am
10396: IPv6 Addressing	Wednesday, March 14th: 9:30am
10398: IPv6 Design	Wednesday, March 14th: 3:00pm
10399: IPv6 Implementation	Wednesday, March 14th: 4:30pm
10400: IPv6 Planning	Wednesday, March 14th: 11:00am
10401: Transitioning to IPv6	Thursday, March 15th: 3:00pm
10413: Common IPv6 Mistakes	Friday, March 16th: 8:00am
10414: IPv6 Deep Dive	Friday, March 16th: 11:00am
10831: IPv6 on z/OS - Part 2	Thursday, March 15th: 9:30am
10836: IPv6 Configuration on z: Hands-on Lab	Thursday, March 15th: 4:30pm

IBM Advanced Technical Skills (ATS) 7

These sessions at SHARE will provide you with valuable and easy-to-digest background information while preparing for your test lab and designing a future production IPv6 environment.

There are also useful web presentations and articles on the same subject. For example, see the youtube broadcast at http://www.youtube.com/watch?v=0oc6mmF_cok

Once you have explored the basics on IPv6, you will have the necessary background to be able to deal with the more detailed information in extensive manuals and in the RFCs. However, you can start your testing without the very extensive knowledge that you think you might need.

Executive Commitment to IPv6 Testing

- Building awareness among executives that the Business Case cannot necessarily rely on Financial Considerations!
  - Usually no way to justify a Return on Investment (ROI) unless the purchase price of exhausted IPv4 addresses becomes an issue
  - Vendors are not usually providing required applications that are IPv6 only
  - Reality of address exhaustion and use of IPv6 by emerging and growth market geographies are the driving force
- Recognition of need to stand behind and provide funding for the IPv6 design and testing
  - Manpower commitments
  - Equipment budget

See the youtube broadcast at http://www.youtube.com/watch?v=0oc6mmF_cok

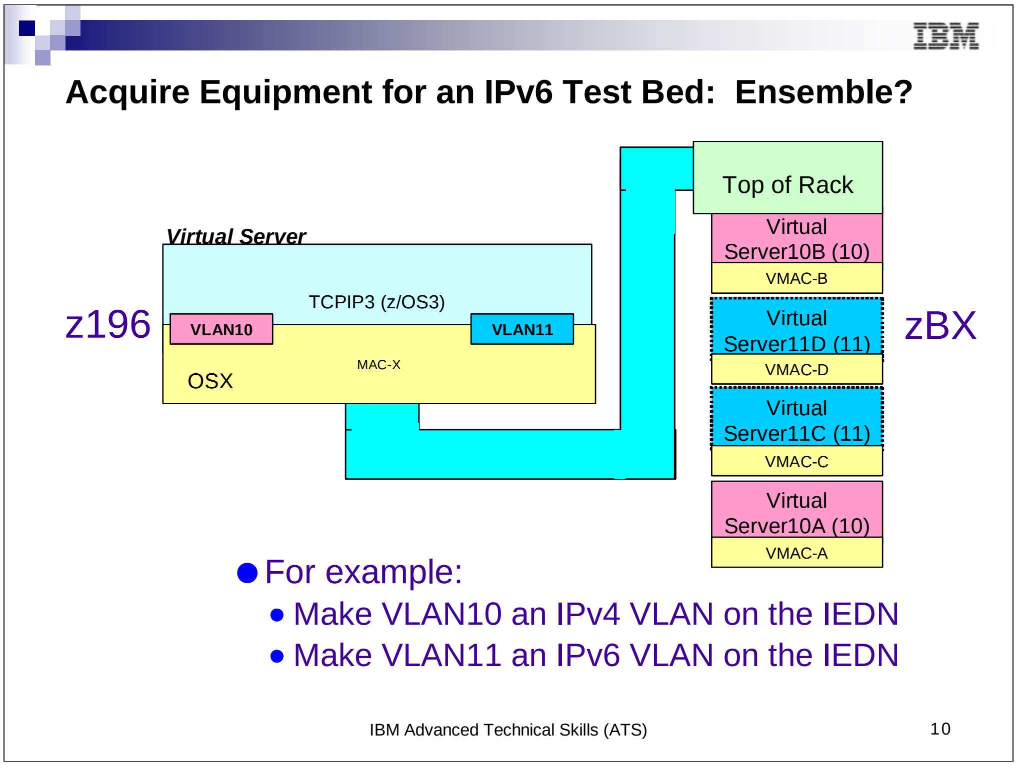
IBM Advanced Technical Skills (ATS) 8

Resorting to the "dark side" (money considerations only) for business justification will probably not work in justifying an IPv6 test bed. Imagination and an understanding of historical reality of IT evolution is necessary! And, if imagination and broad thinking are insufficient to make a compelling case, then perhaps governmental or customer mandates may work their magic.

Excellent information on this subject is available at http://www.youtube.com/watch?v=0oc6mmF_cok

Determine Suitability of Current Environment for IPv6 Testing

- Can I …
  - Carve out a separate HiperSockets network to use for IPv6 testing?
  - Share an existing OSA port for IPv6 testing?
  - Establish a separate LPAR for testing?
  - Establish a separate TCP/IP stack in an existing LPAR for testing?
  - Exploit new z/OS Guests in zVM for testing?
  - Exploit Virtual Servers in an ensemble implementation with zEnterprise and a zBX for testing?
- Are there …
  - Existing routers and switches with IPv6 capability that can collaborate in these plans?



If you are implementing an Ensemble Network with zEnterprise System, you can set up an isolated IPv6 network for test or even for a part of your production network.

The zEnteprise System with its segmentation possiblilities via VLAN gives you the perfect opportunity to test or actually put into production pieces of your network using IPv6 protocols -- IPv6 could be in the Virtual Servers of the z196 node or in the zBX blades or in both -- all separated even from IPv4 by means of separate VLAN IDs, as our visual suggests.


Smarter Systems for a Smarter Planet
IBM
Common Technical Mistakes
IBM Advanced Technical Skills (ATS)
11

Agenda

- **Common Administrative and Organizational Mistakes: Failing to …**
  - Read the simple documentation and planning presentations (web, Share, etc)
  - Gain executive commitment and buy-in to IPv6 testing
  - Acquire a test bed for IPv6 (hardware and software)
  - Create a cross-platform team to discuss testing environment
- **Common Technical Mistakes: Failing to …**
  - Enable IPv6 in z/OS prior to Implementing IPv6
  - Exploit HiperSockets and OSA Enhancements from last releases
    - Convert existing IPv4 QDIO Device/Link coding to INTERFACE coding
    - Exploit HiperSockets zIIP and Multiwrite enhancements
  - Convert Gateway routing statements to Beginroutes/Endroutes routing statements
  - Understand Source IP Address selection for local clients
  - Implement Global Resolver with a Resolver Setup File
  - Implement an IPNodes file for testing and optionally a DNS to handle IPv6
  - Design IPv6 addressing scheme for testing that looks to the future
    - False start: Attempting to use IPv4 numbering in IPv6 addresses
    - False start: “inventing” IPv6 addresses instead of exploiting existing addressing tools
    - If possible, integrating stateless autoconfiguration and DHCP
- **Sample IPv6 Implementation Plan for Test Environment**
  - Including working with cross-platform team to determine the type of addressing prefix that provides the most flexibility for your permanent, non-test deployment of IPv6 addresses
    - Prefix assigned on a geographic basis
    - Prefix assigned by company

IBM Advanced Technical Skills (ATS) 12

IBM

Simple Migration Plan to Prepare for IPv6 Netstat Change

- In preparation for IPv6 Implementation:
  - Enforce policy that all NETSTAT commands be executed in the LONG format to discover effects on automation and operations.
    - "d tcpip,,n,home,format=long" -- or
    - Change IPCONFIG in TCP/IP stack to force FORMAT LONG
    - Create, Execute test plan for this simple change
    - Rewrite any scripts that cannot accommodate FORMAT LONG
- Eliminate use of NETSTAT GATE command
  - Substitute NETSTAT ROUTE for all instances of NETSTAT GATE
  - Create, Execute test plan for this simple change
- Remove reliance of automation on the presence of message identifiers in the TSO version of NETSTAT
- Enable MVS to support IPv6 (Change to hlq.PARMLIB(BPXPRMxx)
  - Create, Execute test plan for this simple change

IBM

Enable IPv6 in z/OS prior to its Implementation

Effect of the IPv6 Addressing on a Participating z/OS Stack

1. Change to BPXPRMxx (UNIX member in PARMLIB)
2. Change to NETSTAT output
   - LONG for IPv6 (or mixed) output (SHORT not supported when IPv6 enabled.)
   - Must use NETSTAT ROUTE to see an IPv6 route
     - and not NETSTAT GATE, which sees only IPv4
   - No Message Identifiers in the LONG Format of _TSO NETSTAT_

LONG Format = IPv6 or IPv4

```
D TCPIP,TCPIPT,N,HOME,FORMAT=LONG
EZD0101I NETSTAT CS V1R10 TCPIPT 034
HOME ADDRESS LIST:
LINKNAME:   VLINK1
  ADDRESS:  192.168.20.102
    FLAGS:  PRIMARY
LINKNAME:   LGIG1F
  ADDRESS:  192.168.20.92
    FLAGS:
…
LINKNAME:   LOOPBACK
  ADDRESS:  127.0.0.1
    FLAGS:
LINKNAME:   LOOPBACK6
  ADDRESS:  ::1
    FLAGS:
7 OF 7 RECORDS DISPLAYED
END OF THE REPORT
```

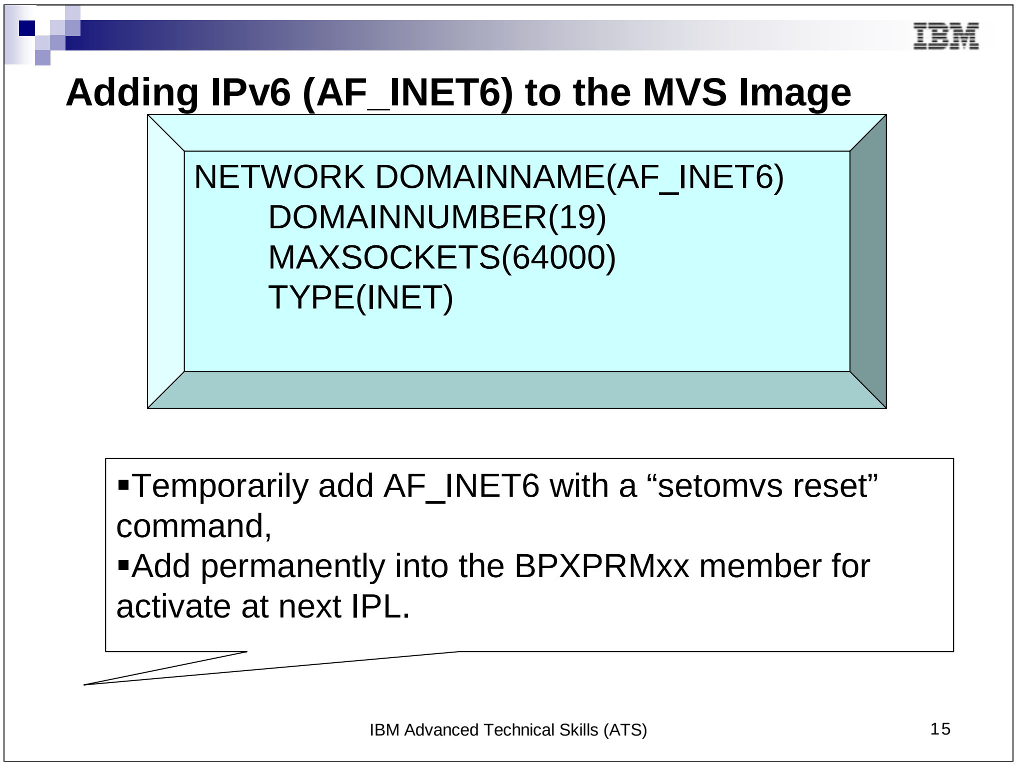
SHORT Format = IPv4 only

```
D TCPIP,TCPIPT,N,HOME
EZZ2500I NETSTAT CS V1R10 TCPIPT 021
HOME ADDRESS LIST:
ADDRESS          LINK             FLG
192.168.20.102   VLINK1           P
192.168.20.92    LGIG1F
10.1.1.2         EZASAMEMVS
…
127.0.0.1        LOOPBACK
6 OF 6 RECORDS DISPLAYED
END OF THE REPORT
```

Otherwise: IPv6 Usage is Transparent

IBM Advanced Technical Skills (ATS) 14

As soon as you add IPv6 to your MVS image, the output for many of your NETSTAT commands changes to the LONG format. In addition you will want to train operators and change automation to use only the NETSTAT ROUTE command and not the NETSTAT GATE command. (NETSTAT ROUTE will display IPv6 addresses; NETSTAT GATE will not.) Finally, the message identifiers that you may have been accustomed to for automation purposes disappear in the LONG format of TSO NETSTAT. Therefore you may wish to change to the MVS console format of the netstat to retain the message identifiers and adjust your automation.



The “setomvs reset” command will add AF_INET6 to the MVS image, but the TCP/IP stack cannot enable it until TCP/IP is recycled. Therefore you could see messages as follows if you use “setomvs reset”:

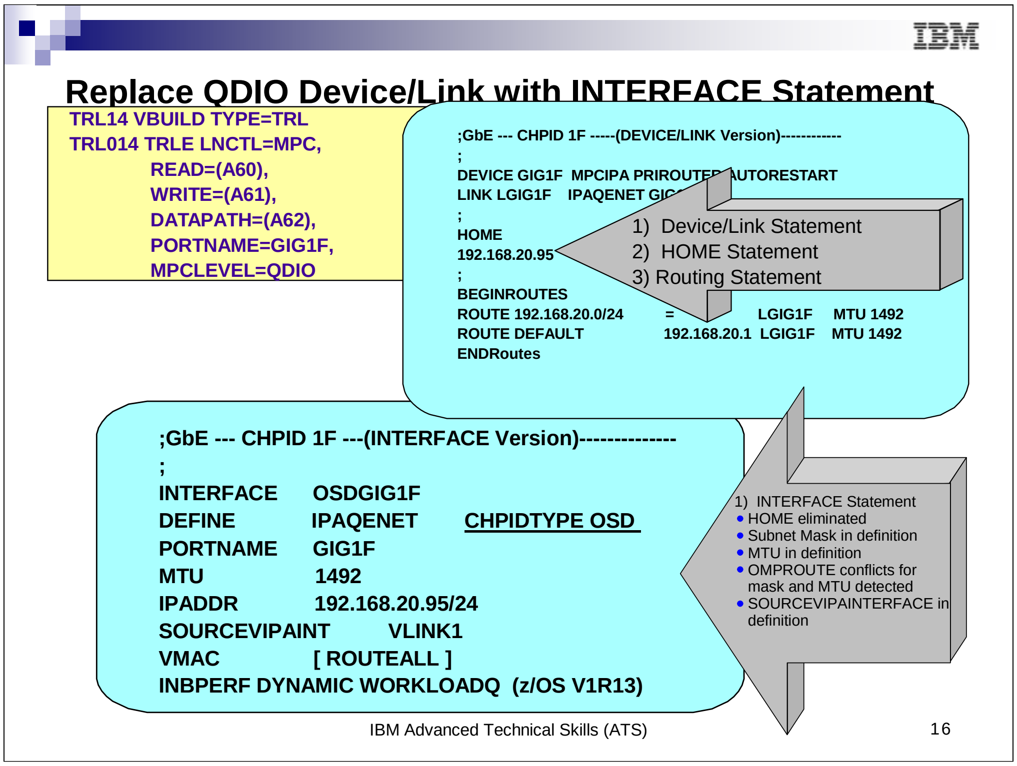
BPXF202I DOMAIN AF_INET6 WAS NOT ACTIVATED FOR FILE SYSTEM TYPE INET. RETURN CODE = 0000045A, REASON CODE = 743A7312

BPXO015I THE SETOMVS COMMAND WAS SUCCESSFUL.

Reason Code of 743A7312 means:

TCPIP JRTCPIP AlreadyInitialized: TCPIP has already initialized

Action: The vfs_network call for AF_INET6 was rejected because TCPIP has already initialized. TCPIP will have to be recycled to process the AF_INET6 network statement.



Convert all OSD (QDIO) Definitions from the old statement syntax of DEVICE and LINK to INTERFACE

Simplifies definitions by including IP address, source VIPAs, MTU sizes, etc.

HOME eliminated:

Uses IPADDR

Subnet Mask in definition

MTU in definition

OMPROUTE conflicts detected for mismatched MTU and Subnet Mask

SOURCEVIPAINTERFACE can be coded directly on the definition where it is needed instead of worrying about the sequence of the HOME list.

Familiarizes you with the new syntax which is REQUIRED for the IPv6 address format. This syntax is also required for IPv4 IEDN OSX interfaces that are implemented with an Ensemble environment for the zEnterprise (z196, z114), the Unified Resource Manager and, optionally, the zBX.

The IP Configuration Guide for your release provides a migration plan and also documents the following TIP:

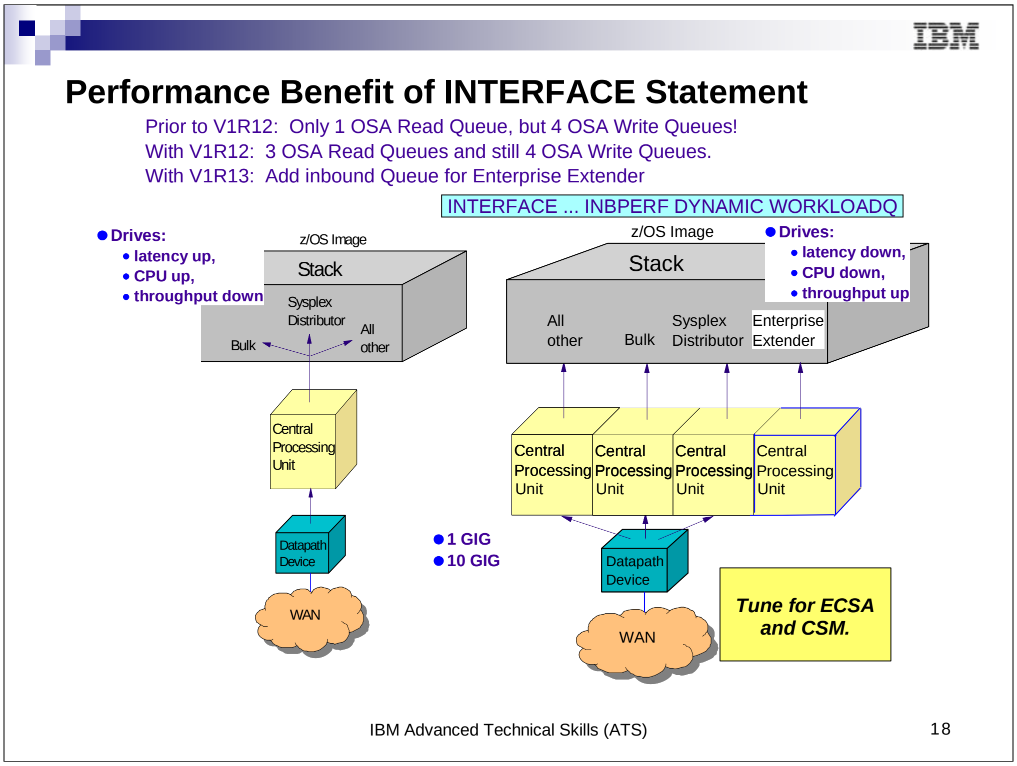
Tip: Optionally, take a dump of the TCP/IP address space and use the CONVERT parameter on the TCPIPCS PROFILE subcommand to display the configuration information at the time of the dump. The resulting output will reflect your IPAQENET DEVICE, LINK, and HOME definitions in INTERFACE statement format, so this might be helpful in converting your profile to use INTERFACE statements. You should thoroughly review the output before you implement any changes. For more information about using the CONVERT parameter on the TCPIPCS PROFILE subcommand, see *z/OS Communications Server: IP Diagnosis Guide*.

Benefits of INTERFACE over Device/Link for IPv4

- Gratuitous ARPs for VIPAs in non-OSA subnet eliminated if Subnet Mask is coded on the IP address
- VIRTUALIZATION of the OSA Port into up to 8 interfaces for IPv4 and 8 for IPv6 in a single Stack with VLAN and VMAC
  - At z/OS V1R13: 32 IPv4 and 32 IPv6 interfaces in single stack
- Default MTU is no longer 576 -- ACTMTU is used
- At V1R11, Optimized Latency Mode on an OSA-E3 takes effect only if coded with INTERFACE
- TCPCONFIG TCPRCVBufrsize 64K
- At V1R12, Inbound Workload Queuing takes effect only if coded with
- INTERFACE
- At V1R12, OSX device is defined only with INTERFACE
- At V1R12, "D OSAINFO" command only displays output with the INTERFACE statement

IBM Advanced Technical Skills (ATS) 17

1. If you define the OSA using DEVICE/LINK statements, then the stack will inform OSA to perform ARP processing for all VIPAs in the home list which can result in numerous unnecessary gratuitous ARPs for VIPAs in an interface takeover scenario.
2. However, if you use the IPv4 INTERFACE statement for IPAQENET, you can control this VIPA ARP processing by configuring a subnet mask for the OSA. If you specify a non-0 num_mask_bits value on the IPADDR parameter of the INTERFACE statement, then the stack will inform OSA to only perform ARP processing for a VIPA if the VIPA is configured in the same subnet as the OSA (as defined by the resulting subnet mask).
3. This is an example of multiple VLAN definitions with two INTERFACE statements for IPAQENET. Each statement defines an IPv4 interface associated with the same OSA-Express port NSQDIO1. Each specifies a subnet mask of 24 bits (‘FFFFFF00”x) and defines a unique subnet.
4. The statements contain different VLAN IDs, and each requests that OSA generate a virtual MAC address (and defaults to ROUTEALL). Each statement specifies the link_name of a static VIPA for the source VIPA function.
5. Because so many definitions that used to reside in the HOME list and in BSDROUTINGPARMS are now included in the INTERFACE definition, it is easier to add and delete interfaces dynamically without having to modify the HOME LIST>
   1. If there is any mismatch between OMPROUTE values (MTU and SUBNET MASK), error messages are generated and the values from OMPROUTE are used.
      - EZZ8163I stack_name MTU value stack_val for interface differs from omproute_procname MTU value omproute_val
      - EZZ8164I stack_name subnet mask value stack_val for interface differs from omproute_procname subnet mask value omproute_val



1. Prior to z/OS V1R12, all inbound QDIO traffic is received on a single read queue regardless of the data type. The maximum amount of storage available for inbound traffic is limited to the read buffer size (64K read SBALs) times the maximum number of read buffers (126). A single process is used to package the data, queue it, and schedule the TCP/IP stack to process it. This same process also performs acceleration functions, such as Sysplex Distributor connection routing accelerator.
2. The TCP/IP stack must separate the traffic types to be forwarded to the appropriate stack component that will process them. For these reasons, z/OS Communications Server is becoming the bottleneck as OSA-Express3 10GbE nears line speed. z/OS Communications Server is injecting latency and increasing processor utilization. This can impede scalability.
3. Under the pre-V1R12 z/OS Communications Server model, another QDIO input process will eventually be driven, and another TCP/IP stack thread, thus allowing multiple threads to process the one inbound read queue. However, this is only done when the OSA detects the host is now "falling behind" using the QDIO interrupt threshold algorithm.
4. z/OS Communications Server is becoming the bottleneck as OSA nears 10GbE line speed, this behavior Injects latency, increases processor utilization, and impedes scalability. For BULK Data, multiple processes are used for inbound traffic when data is accumulating on the read queue. This can cause bulk data packets for a single TCP connection to arrive at the TCP layer out of order. Each time the TCP layer on the receiving side sees out of order data, it transmits a duplicate ACK. Overall, throughput is harmed for bulk data traffic.
5. With z/OS Communications Server V1R12, inbound traffic separation is supported using multiple read queues. TCP/IP will register with OSA which traffic to be received on each read queue. The OSA-Express Data Router function routes traffic to the correct queue.
6. With z/OS Communications Server V1R12, inbound traffic separation is supported using multiple read queues. TCP/IP will register with OSA which traffic to be received on each read queue. The OSA-Express Data Router function routes traffic to the correct queue.
7. Each read queue can be serviced by a separate process. The primary input queue is used for general traffic. One or more ancillary input queues (AIQs) are used for specific traffic types. Sysplex distributor and bulk data traffic is presorted by OSA and routed to z/OS Communications Server on unique AIQs. All other traffic is routed to z/OS Communications Server on the primary input queue. z/OS Communications Server can now process sysplex distributor, bulk data, and other traffic concurrently and independently.
8. The primary queue is always assigned Queue Identifier 1 (QID 1). Each ancillary queue is assigned a Queue Identifier based on when it gets internally registered.
9. The supported traffic types are streaming bulk data and sysplex distributor. Examples of bulk data traffic are FTP, TSM, NFS, and TDMF.
10. Both IP versions are supported for all types of traffic.
    1. With bulk data traffic separated onto its own read queue, TCP/IP will service the bulk data queue from a single processor. This solves the out of order delivery issue – there are no more race conditions.
    2. With sysplex distributor traffic separated onto its own read queue, it can be efficiently accelerated or presented to the target application.
    3. All other traffic is processed simultaneous with the bulk data and sysplex distributor traffic
    4. The dynamic LAN idle timer is updated independently for each read queue. This ensures the most efficient processing of inbound traffic based on the traffic type.
11. The QDIO inbound workload queuing function is enabled with the INBPERF DYNAMIC WORKLOADQ setting on IPAQENET and IPAQENET6 INTERFACE statements. WORKLOADQ is not supported for INBPERF DYNAMIC on IPAQENET LINK statements. WORKLOADQ does require the VMAC on the INTERFACE definition, but you can allow just a dynamically generated value for VMAC. For steps to convert from IPv4 IPAQENET DEVICE, LINK, and HOME definitions to the IPv4 IPAQENET INTERFACE statement refer to z/OS Communications Server: IP Configuration Guide.
12. Each ancillary queue will consume:
    1. Approximately nine additional pages of ECSA
    2. An additional but tunable amount of fixed CSM data space as specified by the READSTORAGE parameter

Diagnostic Benefits of Migration to INTERFACE: OSAINFO Command

```
EZZ0053I COMMAND DISPLAY TCPIP,,OSAINFO COMPLETED SUCCESSFULLY
EZD0031I TCP/IP CS V1R12  TCPIP Name: TCPSVT    15:39:52
Display OSAINFO results for IntfName: V6O3ETHG0
PortName: O3ETHG0P  PortNum: 00  Datapath: 2D64   RealAddr: 0004
PCHID: 0270         CHPID: D6    CHPID Type: OSD  OSA code level: 5D76
Gen: OSA-E3         Active speed/mode: 10 gigabit full duplex
Media: Singlemode Fiber          Jumbo frames: Yes  Isolate: No
PhysicalMACAddr: 001A643B887C    LocallyCfgMACAddr: 000000000000
Queues defined Out: 4  In: 3     Ancillary queues in use: 2
Connection Mode: Layer 3         IPv4: No    IPv6: Yes
SAPSup: 00010293                 SAPEna: 00010293

IPv6 attributes:
  VLAN ID:   12             VMAC Active: Yes
  VMAC Addr: 0206100B2068   VMAC Origin: Cfg      VMAC Router: All
  AsstParmsEna: 00215C60    OutCkSumEna: 00000000  InCkSumEna: 00000000
```

OSD, OSX, or OSM

IBM Advanced Technical Skills (ATS) 19

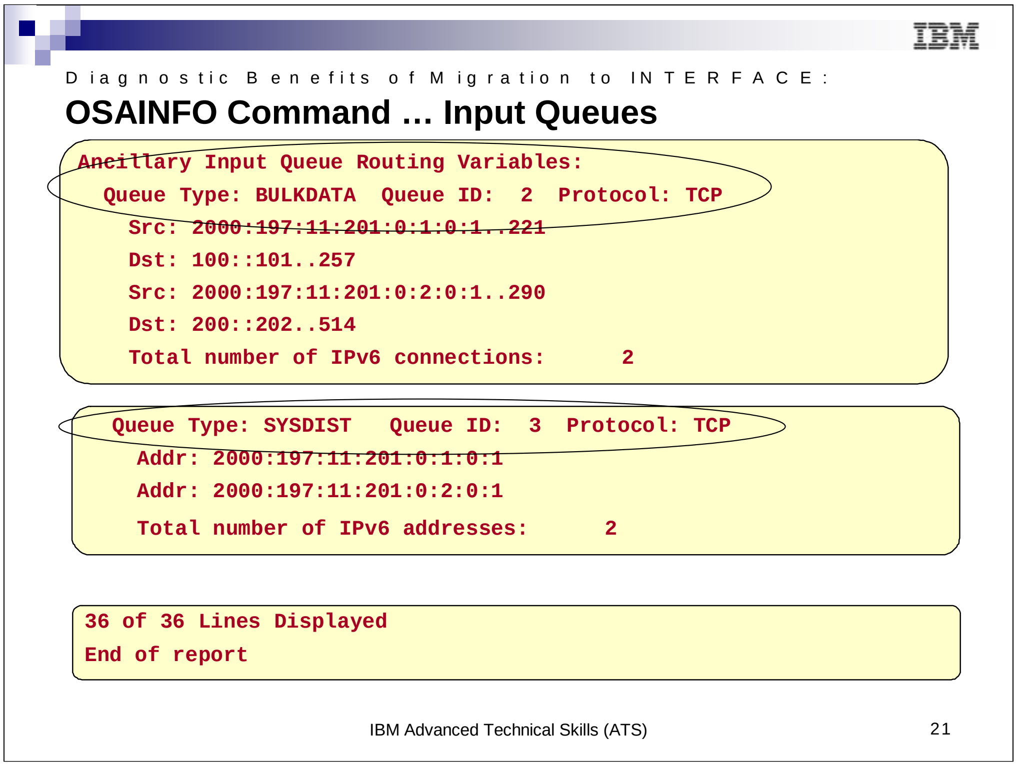
1. D OSAINFO is valid on an OSA-E3 in QDIO Mode (either CHPID Type of OSD, OSX, or OSM) as long as the interface has been defined with the INTERFACE Statement.
2. OSA requirements:
    1. OSA-Express3 Ethernet features in QDIO mode running on an
    2. IBM System z10
    3. See the 2097DEVICE and 2098DEVICE Preventive Service Planning (PSP) buckets for the required MCL levels
3. You can issue the DISPLAY OSAINFO command to determine if OSA supports the command
    1. INTFNAME must be defined as IPAQENET or IPAQENET6
    2. INTFNAME must be active
4. The command sorts addresses and ports in ascending order
5. Impact of command on both OSA and Communications Server resources should be insignificant
6. OSA Requirements:
    1. OSA-Express3 Ethernet features in QDIO mode running on an IBM System z10
    2. See the 2097DEVICE and 2098DEVICE Preventive Service Planning (PSP) buckets for the required MCL levels
7. Sections of the Output Display:
    1. This part of the sample reply is the start of the BASE section. The BASE section shows general information about the OSA such as the CHPID (in this sample the CHPID is D6).
        1. All of the fields displayed in the reply are documented in z/OS Communications Server IP System Administrator's Commands Version 1 Release 12.
        2. Message EZZ0053I is not part of the report but instead it's issued when the display command is accepted.
        3. Message EZD0031I is the 1st message in the multi-write to operator reply and is issued when all information has been received from OSA
    2. This part of the sample reply is the end of the BASE section. This sample shows information about the IPv6 Layer 3 attributes such as the Global VLAN ID and VMAC information.
        1. If the data device has IPv4 enabled (which this sample does not), the IPv4 Layer 3 attributes are displayed.
        2. If the data device has IPv6 enabled (which this sample does), the IPv6 Layer 3 attributes are displayed.
    3. If the data device has IPv4 and IPv6 enabled , the IPv4 Layer 3 attributes are displayed first, followed by the IPv6 Layer 3 attributes.

Diagnostic Benefits of Migration to INTERFACE: OSAINFO Command … Address Table display

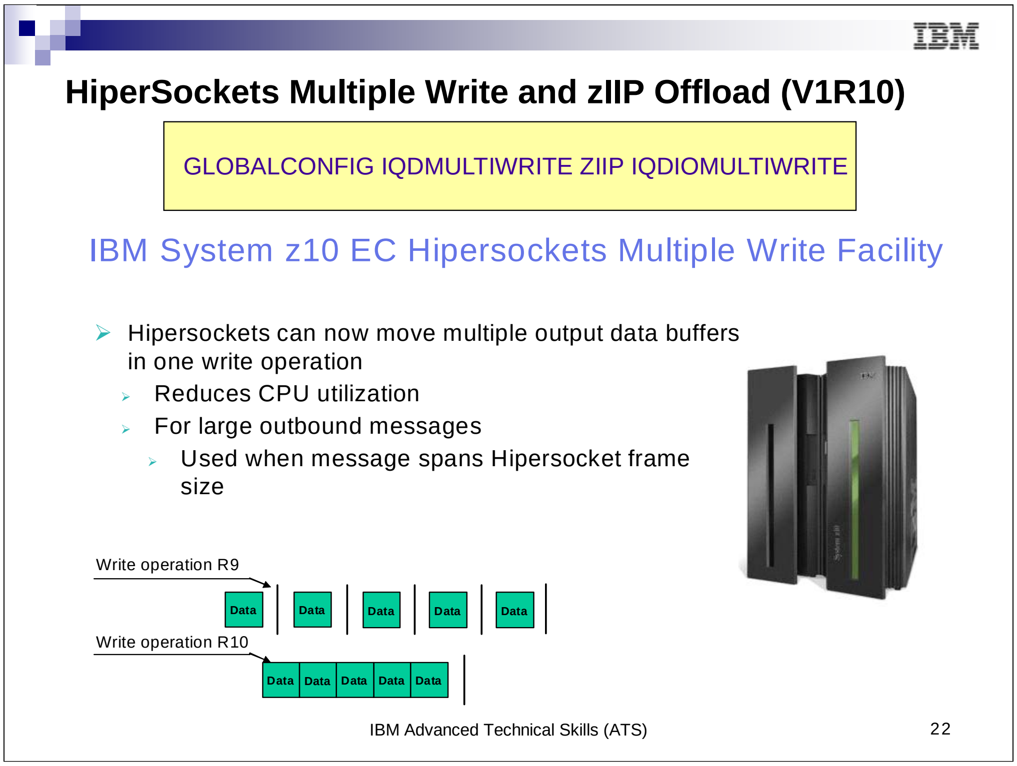
```
Registered Addresses:
  IPv4 Unicast Addresses:
    ARP: Yes  Addr: 16.2.16.107
    Total number of IPv4 addresses:      1
  IPv4 Multicast Addresses:
    MAC: 01005E000001  Addr: 224.0.0.1
    Total number of IPv4 addresses:      1
  IPv6 Unicast Addresses:
    Addr: FE80::11:16:32:104
    Total number of IPv6 addresses:      1
  IPv6 Multicast Addresses:
    MAC: 333300000001  Addr: FF02::1
    MAC: 3333FF010001  Addr: FF02::1:FF01:1
    MAC: 3333FF010002  Addr: FF02::1:FF01:2
    MAC: 3333FF010003  Addr: FF02::1:FF01:3
    Total number of IPv6 addresses:      4
```

IBM Advanced Technical Skills (ATS) 20

1. This part of the sample reply is the REGADDRS section in its entirety. Displayed here are all the IPv4 and IPv6 unicast and multicast addresses registered with the OSA.
   1. Note that the IPv4 information conflicts with other sections of the reply. The IPv4 information was inserted here for illustration purposes only.
   2. If the interface has IPv4 enabled (which this sample does), the IPv4 registered unicast and multicast addresses are displayed. The ARP field indicates if the OSA is performing ARP for an IPv4 unicast address.
   3. If the interface has IPv6 enabled (which this sample does), the IPv6 registered unicast and multicast addresses are displayed.
2. Continued sections of the Output display:



1. This part of the sample reply is the REGADDRS section in its entirety. Displayed here are all the IPv4 and IPv6 unicast and multicast addresses registered with the OSA.
2. Note that the IPv4 information conflicts with other sections of the reply. The IPv4 information was inserted here for illustration purposes only.
3. If the interface has IPv4 enabled (which this sample does), the IPv4 registered unicast and multicast addresses are displayed. The ARP field indicates if the OSA is performing ARP for an IPv4 unicast address.
4. If the interface has IPv6 enabled (which this sample does), the IPv6 registered unicast and multicast addresses are displayed.
5. Continued sections of the Output display:
    1. BULKDATA:
        1. This part of the sample reply is the BULKDATA section in its entirety. Displayed here are the source and destination IP address and ports of the TCP connections for which OSA is performing QDIO Inbound Workload Queuing for streaming connections. If the interface has QDIO Inbound Workload Queuing enabled for BULKDATA and there is at least one connection, the BULKDATA section is displayed.
        2. Note that you can see IPv4 or IPv6 addresses here but not both as QDIO Inbound Workload Queuing is not allowed when a single datapath device is used for both IPv4 and IPv6.
    2. Sysplex Distribution:
        1. This part of the sample reply is the SYSDIST section in its entirety. Displayed here are the destination IP address for which OSA is performing QDIO Inbound Workload Queuing for sysplex distributor. If the interface has QDIO Inbound Workload Queuing enabled for sysplex distributor and at least one destination address, the SYSDIST section is displayed.
        2. Note that you can see IPv4 or IPv6 addresses here but not both as Inbound Workload Queuing is not allowed when a single datapath device is used for both IPv4 and IPv6.
6. The first number shows the total number of lines displayed. The second number shows the total number of lines it's possible to display. The MAX operator can be specified to limit the total number of lines displayed.
7. If MAX=* is specified and more than 65,535 lines are required, Report truncated: Max lines limit reached is displayed instead of the message with the counts.



1. Starting with the IBM System z10 includes a function called Hipersockets Multiple Write was introduced. This allows multiple data buffers to be moved from one system image to another across Hipersockets with one operation. This can reduce CPU utilization.
2. With HiperSockets Multiple Write enabled you should see a performance improvement and reduction in CPU utilization for large outbound messages.
   1. .. zIIP assist will also help reduce costs associated with general CPU utilization.
   2. Valid for .. Both HiperSockets Multiple Write and zIIP-Assisted HiperSockets
   3. Multiple Write are disabled by default. Enable them using the new options on the GLOBALCONFIG statement.
   4. .. There are no WLM (enclave) configuration changes required.
   5. .. The PROJECTCPU function in z/OS Workload Manager can be used to project zIIP effectiveness.
3. When enabled, HiperSockets Multiple Write will be used anytime a message spans the Hipersockets frame size, thus requiring multiple output buffers to transfer the message. Therefore, it will only be used for larger outbound messages. Spanning multiple output data buffers can be affected by a number of factors including:
   1. Hipersocket frame size
   2. Application socket send size
   3. TCP send size
   4. MTU size
4. SUMMARY: HiperSockets Multiple Write
   1. Requirements
      1. • IBM System z10 or higher
   2. Restrictions
      1. • Unsupported if z/OS is running as a guest in a z/VM environment.
      2. • Supported for large outbound messages only
5. SUMMARY: .. zIIP-Assisted HiperSockets Multiple Write
   1. Requirements
      1. • HiperSockets Multiple Write must be enabled
   2. Restrictions
      1. • Will only be used for large outbound TCP messages (that originate in this host).

Replace Static Gateway routes with BEGinroutes/Endroutes

```
BEGINRoutes
;      Destination Subnet Mask      First Hop    Link Name Packet Size
;
 ROUTE FD49:DC08:87BD:A814/64      =            IP6GIG2A    MTU 1492  REPL
 ROUTE FD49:DC08:87BD:A815/64      =            IP6GIG2B    MTU 1492  REPL
 ROUTE FD49:DC08:87BD:A816/64      =            IP6IQ2DE    MTU 8992  REPL
 ROUTE DEFAULT                 192.168.20.1     OSDGIG1F  MTU 1492
 ROUTE DEFAULT6 FE80::1:2:601                   IP6GIG2A MTU 1492
 ROUTE DEFAULT6 FE80::1:2:603                   IP6IQ2DE MTU 1492
ENDRoutes
```

The old GATEWAY statement does not support IPv6 addresses. It has also been the cause of many misunderstandings and misconfigurations due to its inconsistent syntax that doesn't match anything else on other platforms. BEGINROUTES is the recommended statement for static route definitions ever since OS/390 V2R10. It supports both IPv4 and IPv6 addresses.

IBM Advanced Technical Skills (ATS) 23

BeginRoutes and GATEWAY cannot coexist in the same profile. The first one encountered will be used.

BeginRoutes allows you to specify the netmask as a number of significant bits in the netmask.

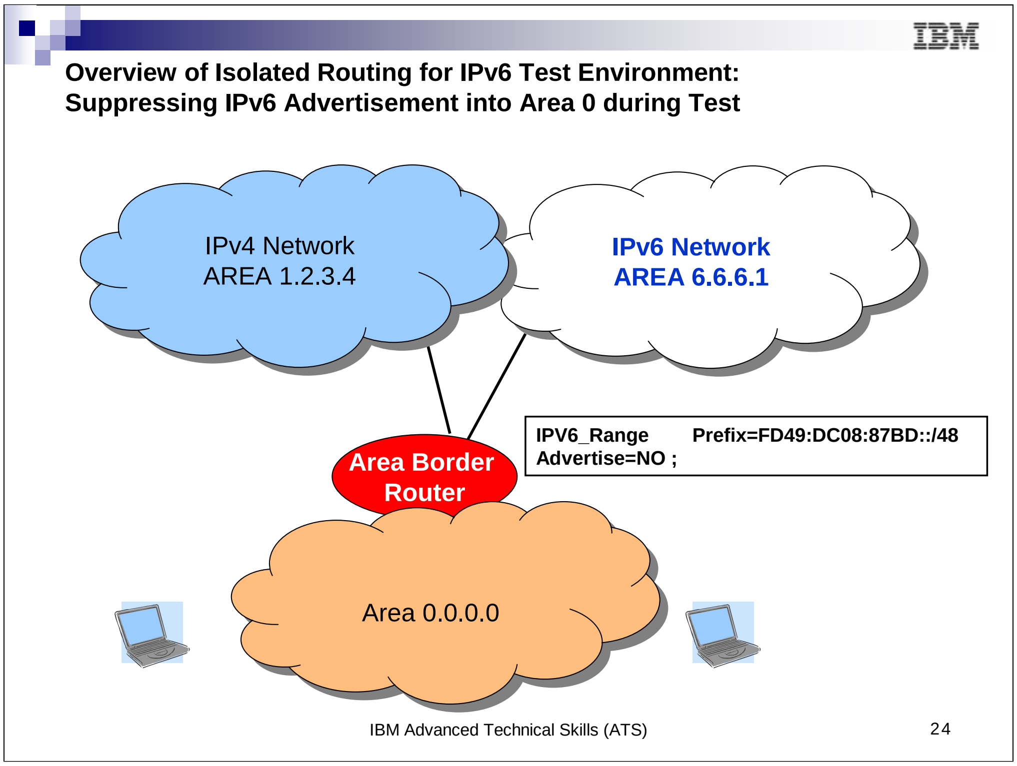
All static routes to the same destination are considered equal cost. There is no limit to the number of equal cost multipath routes you can specify.

You must have a direct route to the first hop.

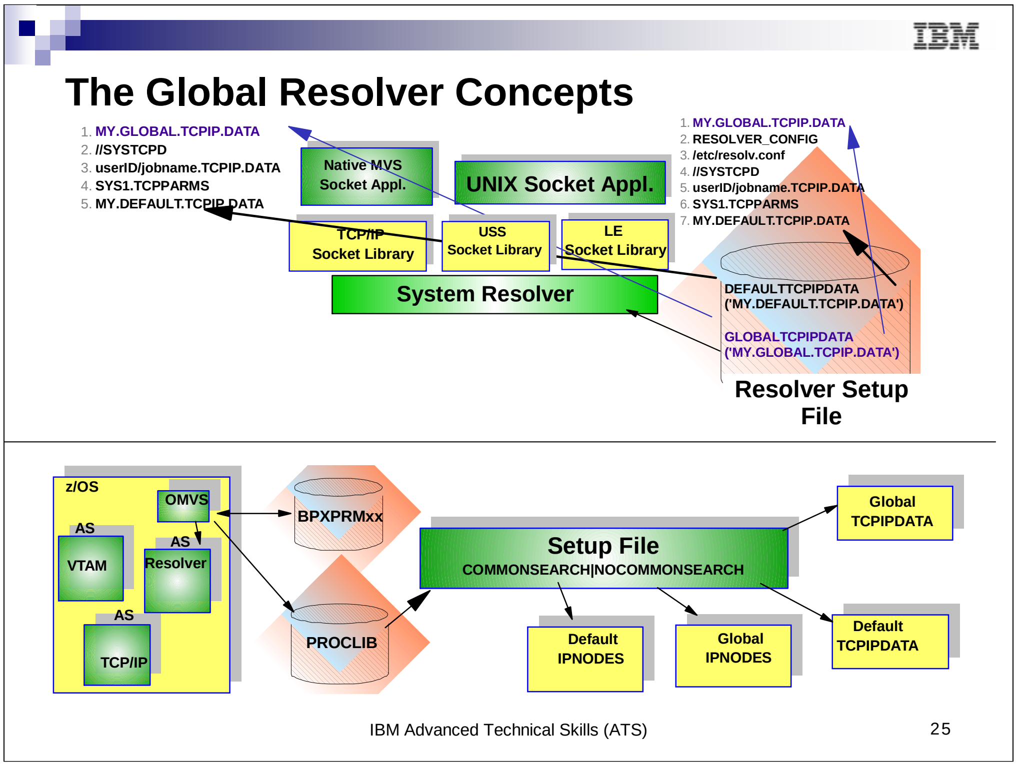
VIPA links are not allowed on GATEWAY or BeginRoute statements

Advantages to using BeginRoutes:

- can specify network/subnet routes for multicast addresses
- simplified syntax, especially for supernet routes
- supports both IPv6 and IPv4 addresses
- Allows options not possible on GATEWAY statements; is the only form of static routing in z/OS that will continue to enjoy enhancements.



Until ready to begin advertising the IPv6 range into Area 0, code the Area Border Router to suppress the advertisement of the IPv6 ranges



The top half of the visual shows you the benefits of the Global Resolver – which allows the use of a single set of resolver files for both UNIX and MVS applications, and, introduces an IPNodes file that allows you to configure both IPv6 and IPv4 addresses for name resolution. (The host local file only permits IPv4 addresses.)

The bottom half of the visual shows you how a customized SETUP file points to the important resolver files like TCPdata and IPNodes, etc.

Sample Resolver Setup File

Modified from SYS1.TCPIP.SEZAINST(RESSETUP)

```
;
DEFAULTTCPIPDATA('TCPIP.TCPIP.DATA')
;
;GLOBALTCPIPDATA('TCPCS.SYS.TCPPARMS(GLOBAL)')
; GLOBALTCPIPDATA(/etc/tcpipglobal.data)
;
;GLOBALIPNODES('TCPCS.SYS.TCPPARMS(IPNODES)')
; GLOBALIPNODES('TCPCS.ETC.IPNODES')
; GLOBALIPNODES(/etc/ipnodes)
;
;DEFAULTIPNODES('TCPCS.SYS.TCPPARMS(IPNODES)')
; DEFAULTIPNODES('TCPCS.ETC.IPNODES')
; DEFAULTIPNODES(/etc/ipnodes)
;
NOCOMMONSEARCH
; COMMONSEARCH
```

Recommend "COMMONSEARCH" & not this Default!

IBM Advanced Technical Skills (ATS) 26

Since V1R2, the System Resolver is a separate address space. The address space is automatically started by OMVS. Its purpose is to initialize the resolver API facilities and also to collect resolver CTRACE debugging information. Similarly to the VTAM and TCPIP APIs, execution of the resolver API runs in the application address space which is requesting the services.

The system resolver address space needs to be active in order for any name-to-IP address or IP address-to-name resolution to take place. This includes usage of DNSs as well as local "hard" files. Likewise non-resolver TCPIP.DATA information (e.g., DatasetPrefix, Hostname and TCPIP socket API tracing settings) also requires the system resolver.

If no customization is done the V2R10 search order continues to be used to locate the TCPIP.DATA file parameters, local hosts files, etc..

V1R4 adds the ability to have a new local hosts file which may have both IPv4 and IPv6 information

V1R4 adds new statements for the setup file -- the COMMONSEARCH or NOCOMMONSEARCH statement as well as pointers to LOCAL HOSTs files called IPNODES which contain IPv6 and/or IPv4 addresses.

Sample IPNodes File with IPv4 and IPv6

```
;Entries in the IPNODES file have the following format:
;
;  Address HostName
;  Address HostName1 HostName2 HostName3 ..... HostName35
;
;  Address: is an IP address, it can be IPV4 or IPV6 address.
;          Note: IPv4-mapped IPv6 address is not allowed.
;
;  HostName: the length of the hostname is up to 128 characters,
;           and each IP address can have up to 35 hostnames.
;
   9.67.43.100 NAMESERVER
   9.67.43.126 RALEIGH
   9.67.43.222 HOSTNAME1.RALEIGH.IBM.COM
   129.34.128.245 YORKTOWN WATSON
   1::2 TESTIPV6ADDRESS1
   1:2:3:4:5:6:7:8 TESTIPV6ADDRESS2
;
```

IBM Advanced Technical Skills (ATS) 27

This file is pointed to in the Resolver Setup file. Notice that it can contain both IPv4 and IPv6 addresses.

If both IPv6 and IPv4 addresses are associated with the same DNS name for a connection or association request, the IPv6 addresses are preferred and sorted ahead of the IPv4 addresses. Therefore, it is wisest to assign separate DNS or IPNodes names to the IPv6 addresses from the DNS names assigned to the IPv4 addresses – especially during testing. In order to test what happens when both IPv6 and IPv4 addresses are returned during a resolver lookup, you might even assign a third DNS name.

Implement Global Resolver and its Setup File in TCP/IP

Sample Proc in "SYS1.TCPIP.SEZAINST(TCPIPROC)"

```
//TCPIP    PROC PARMS='CTRACE(CTIEZB00),IDS=00'
//*
//TCPIP    EXEC PGM=EZBTCPIP,REGION=0M,TIME=1440,
//              PARM='&PARMS'
//*
//*
//*
//*
//*
//*
//*
//*
//PROFILE  DD DISP=SHR,
//*             DSN=TCPIVP.TCPPARMS(PROFILE)
//*
//* SYSTCPD explicitly identifies which data set is to be
//* used to obtain the parameters defined by TCPIP.DATA
//* when no GLOBALTCPIPDATA statement is co
//* (or when a second TCPIPDATA is to be co
//* a GLOBALTCPIPDATA -- note from GJD).
//*
//*SYSTCPD  DD DISP=SHR,
//*            DSN=TCPIVP.TCPPARMS(TCPDATA)
//*
```

//PROFILE DD DSN=TCPIVP.TCPPARMS(PROFILE)

SEARCH ORDER (PROFILE):

- //PROFILE DD DSN=aaa.bbb.ccc(anyname)
- jobname.nodename.TCPIP
- hlq.nodename.TCPIP
- jobname.PROFILE.TCPIP
- TCPIP.PROFILE.TCPIP

//PROFILE DD DSN=TCPIVP.TCPPARMS(TCPDATA)

SEARCH ORDER (TCPDATA):

- If defined in Resolver Setup, Use GLOBALTCPIPDATA
- OPTIONAL: 2nd SYSTCPD as found by Resolver Search Order; Example is: //SYSTCPD DD DSN=aaa.bbb.ccc(anyname)

IBM Advanced Technical Skills (ATS) 28

When you implement a Global Resolver you need not point to a separate TCP data in your JCL procedures unless you need discrete entries in that file for certain JCL. However, beware … examine the guidelines in the IP Configuration Guide and IP Configuration Reference for rules governing what MUST be in the Global TCP data file and what can be in the optional secondary TCP data file.

Learn about Source IP Address Selection

1. Sendmsg( ) using the IPV6_PKTINFO ancillary option specifying a nonzero source address (RAW and UDP sockets only)
2. Setsockopt( ) IPV6_PKTINFO option specifying a nonzero source address (RAW and UDP sockets only)
3. Explicit bind to a specific local IP address
4. bind2addrsel socket function (AF_INET6 sockets only)
5. PORT profile statement with the BIND parameter
6. SRCIP profile statement (TCP connections only)
7. TCPSTACKSOURCEVIPA parameter on the IPCONFIG or IPCONFIG6 profile statement (TCP connections only)
8. SOURCEVIPA: Static VIPA address from the HOME list or from the SOURCEVIPAINTERFACE parameter
9. HOME IP address of the link over which the packet is sent

TCP/IP determines the source IP address for a TCP outbound connection, or for a UDP or RAW outbound packet, using the sequence you see above, listed in descending order of priority.

Some of the newer and simplest ways of assigning a source ip address are: SRCIP statement for TCP only (option 6 above) and SOURCEVIPAINTERFACE (option 8 above).

If both IPv6 and IPv4 addresses are associated with the same DNS name for a connection or association request, the IPv6 addresses are preferred and sorted ahead of the IPv4 addresses. Therefore, it is wisest to assign separate DNS or IPNodes names to the IPv6 addresses from the DNS names assigned to the IPv4 addresses – especially during testing. In order to test what happens when both IPv6 and IPv4 addresses are returned during a resolver lookup, you might even assign a third DNS name.

For a TCP connection, the source address is selected for the initial outbound packet, and the same source IP address is used for the life of the connection. For the UDP and RAW protocols, a source IP address selection is made for each outbound packet. Please consult the IP Configuration Guide for more details.

IBM Advanced Technical Skills (ATS) 29

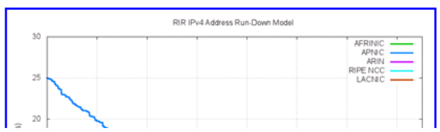
Start Testing Now! IPv4 Address Exhaustion!

http://www.sixxs.net/tools/grh/ula/
Generated 15 March 2012

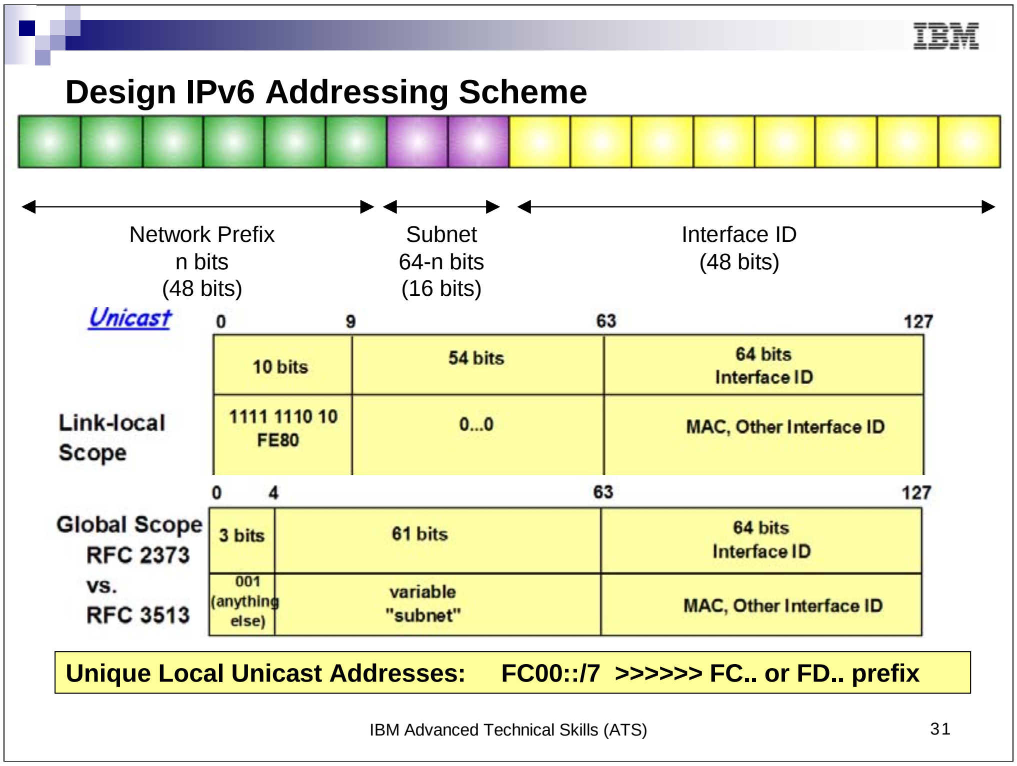
IANA Unallocated Address Pool Exhaustion:
03-Feb-2011

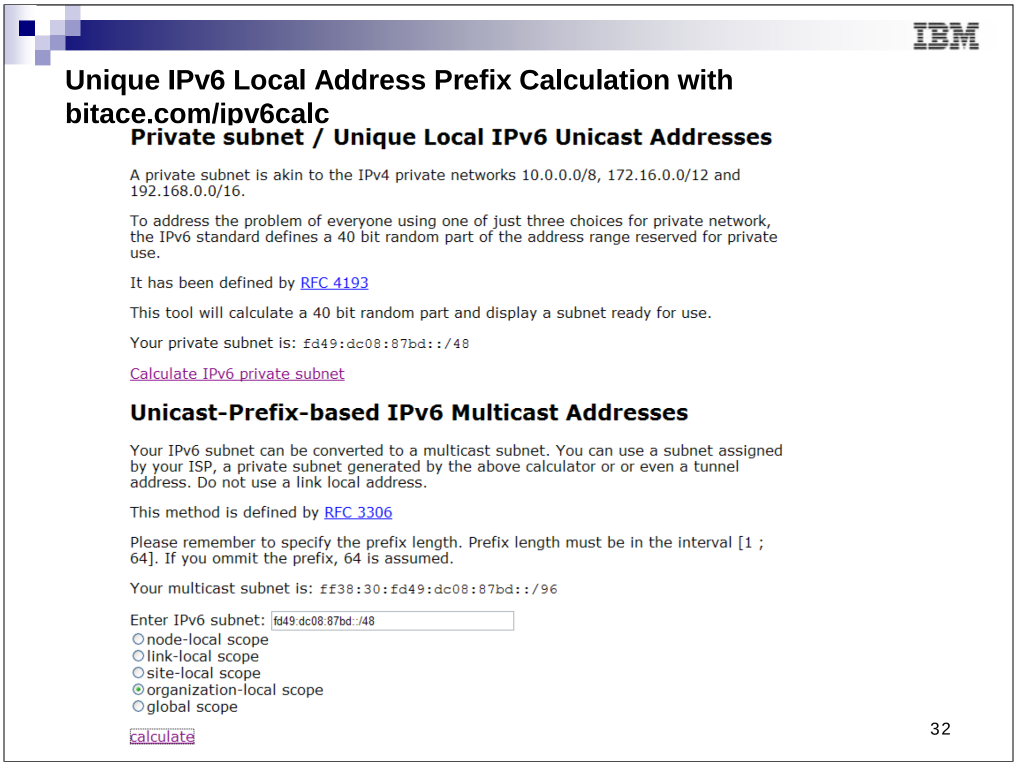
Projected RIR Address Pool Exhaustion Dates:

RIR	Projected Exhaustion Date	Remaining Addresses in RIR Pool (/8s)
APNIC:	**19-Apr-2011**	1.1664
RIPENCC:	**08-Aug-2012**	2.7082
ARIN:	**27-Jul-2013**	5.5680
LACNIC:	**28-Jan-2014**	3.8490
AFRINIC:	**29-Oct-2014**	4.3355



IBM Advanced Technical Skills (ATS) 30


IBM
Design IPv6 Addressing Scheme
Network Prefix
n bits
(48 bits)
Subnet
64-n bits
(16 bits)
Interface ID
(48 bits)
Unicast
0
9
63
127
10 bits
54 bits
64 bits
Interface ID
Link-local
Scope
1111 1110 10
FE80
0...0
MAC, Other Interface ID
0
4
63
127
Global Scope
RFC 2373
vs.
RFC 3513
3 bits
61 bits
64 bits
Interface ID
001
(anything
else)
variable
"subnet"
MAC, Other Interface ID
Unique Local Unicast Addresses: FC00::/7 >>>>>> FC.. or FD.. prefix
IBM Advanced Technical Skills (ATS)
31



Multiple tools exist on the internet to generate unique local unicast addresses. Bitace.com/ipv6calc is one such tool; http://www.sixxs.net/tools/grh/ula/ is another. A simple web search will uncover many others. Some tools are easier to use than others and you should look at several to determine which is easiest to use.

Should I Model an Interface ID after an IPv4 Address?

- Sample IPv4 Address:
  - 192.168.20.94

- Sample IPv6 Address:
  - fd49:dc08:87bd:0000:*192*:168:20:94/48
  - 0……………………63 64…………128 bits
    - 192 resolves to 0000 0001 1001 0010
      - Bit 70 = 0
      - Bit 71 = 1

❖You cannot always successfully model an IPv6 address after an IPv4 address because fields of the new address might cause the address to be interpreted as a "reserved IPv6 address."

❖**Problem lies in the value of bits 71 and 72.**

IBM Advanced Technical Skills (ATS) 33

Error Message at z/OS with Reserved IPv6 Address

EZZ0726I RESERVED IPv6 ADDRESS OF ipv6addr ON LINE lineno CANNOT BE SPECIFIED

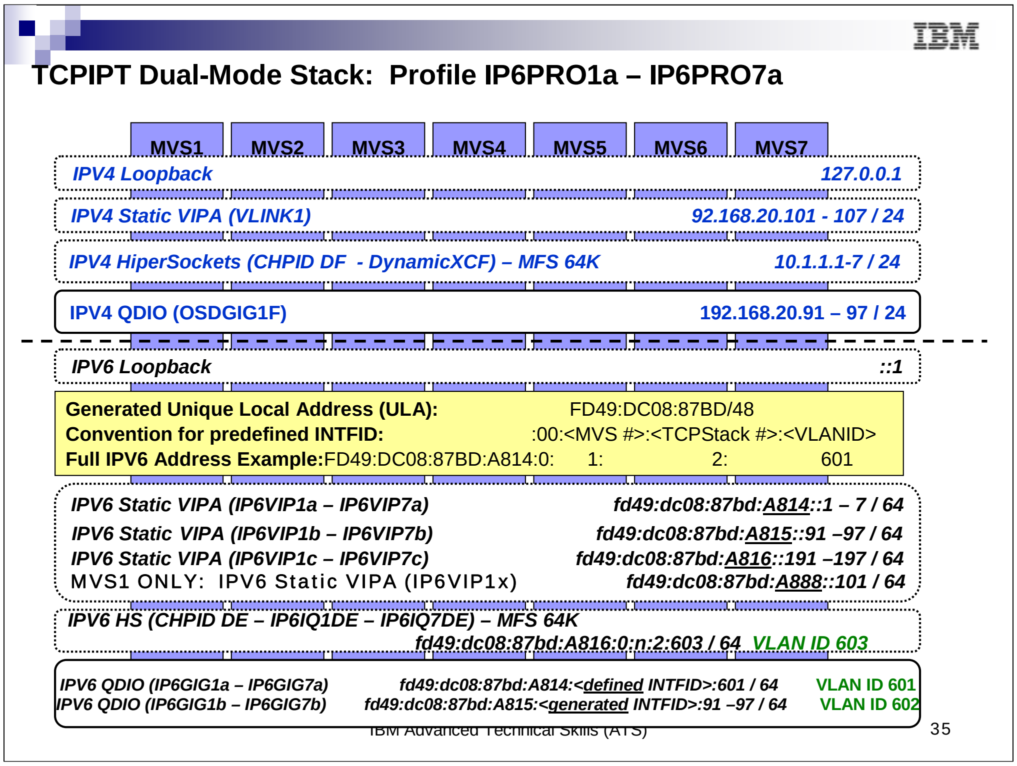
Explanation:
A reserved IPv6 address cannot be specified on the IP configuration statement at line lineno.

A reserved IPv6 address is one of the following:

Universal/local bit is set (bit 71).
Individual/group bit is set (bit 72).

The first four bytes of the interface ID are X'00005EFE'.
The first 57 bits of interface ID, except universal/local and individual/group bits, are set to 1 (for example, FCFFFFFFFFFFFFF8/57).

See the z/OS Communications Server: IPv6 Network and Application Design Guide for information about IPv6 addresses.



This is a sample IP addressing scheme that was used to build a test lab. We generated a 48-bit prefix for our “company” by using one of the many web-based generator tools for Unique Local Unicast Addresses (ULAs). Then we used the next 16 bits to assign subnet “addresses” for individual network types: HiperSockets, vs. LAN attachments, vs. Static VIPA networks, and so on. Thus, the entire subnet mask for each type of network becomes “/64.”

For the first phase of our testing plan we generally avoided the use of generated Interface IDs for the OSA LAN ports and networks. Instead we defined the interface IDs. Our convention for the predefined INTFID worked for this test lab; you need to work with a group of your colleagues to come up with an INTFID numbering scheme that is valid for your company’s test network. Our convention was:

:00:<MVS #>:<TCPStack #>:<VLANID>

Smarter Systems for a Smarter Planet

Sample IPv6 Implementation Plan for Test Environment

IBM Advanced Technical Skills (ATS) 36

IBM

Agenda

- **Common Administrative and Organizational Mistakes: Failing to …**
  - Read the simple documentation and planning presentations (web, Share, etc)
  - Gain executive commitment and buy-in to IPv6 testing
  - Acquire a test bed for IPv6 (hardware and software)
  - Create a cross-platform team to discuss testing environment
- **Common Technical Mistakes: Failing to …**
  - Enable IPv6 in z/OS prior to Implementing IPv6
  - Exploit HiperSockets and OSA Enhancements from last releases
    - Convert existing IPv4 QDIO Device/Link coding to INTERFACE coding
    - Exploit HiperSockets zIIP and Multiwrite enhancements
  - Convert Gateway routing statements to Beginroutes/Endroutes routing statements
  - Understand Source IP Address selection for local clients
  - Implement Global Resolver with a Resolver Setup File
  - Implement an IPNodes file for testing and optionally a DNS to handle IPv6
  - Design IPv6 addressing scheme for testing that looks to the future
    - False start: Attempting to use IPv4 numbering in IPv6 addresses
    - False start: “inventing” IPv6 addresses instead of exploiting existing addressing tools
    - If possible, integrating stateless autoconfiguration and DHCP
- **Sample IPv6 Implementation Plan for Test Environment**
  - Including working with cross-platform team to determine the type of addressing prefix that provides the most flexibility for your permanent, non-test deployment of IPv6 addresses
    - Prefix assigned on a geographic basis
    - Prefix assigned by company

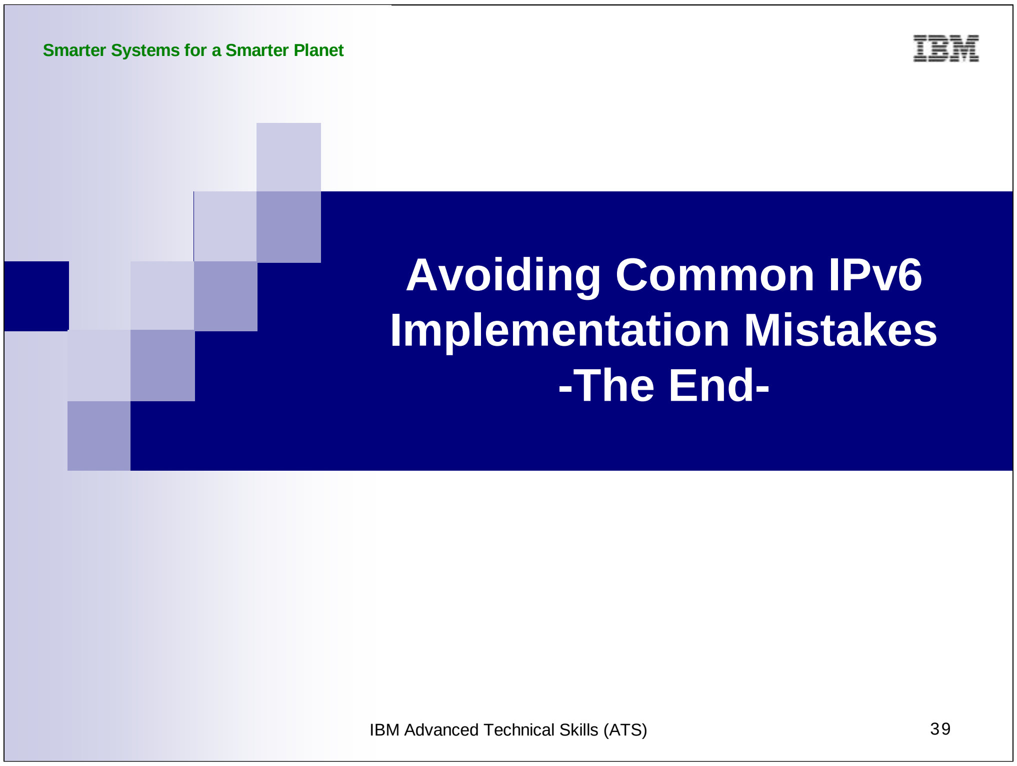
IBM Advanced Technical Skills (ATS) 37

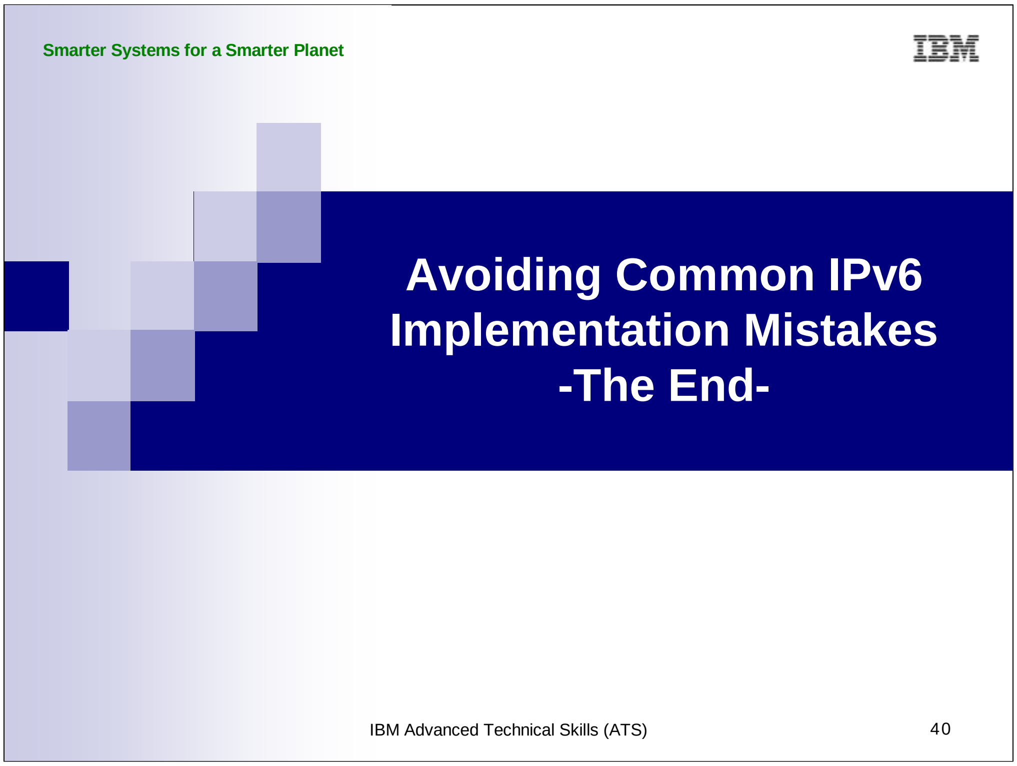
Implementation Plan for Test

- Arrange for the LPARs and Equipment that will be used for testing
- Exploit features of an IPv4 network that will facilitate migration to IPv6
  - Interface definitions for IPv4
  - IPNodes file
  - Etc.
- Determine Testing Stages and Personnel:
  - 1. Simple Design Stage
    - Predefined Addresses
    - IPNodes File for Name-to-Address Resolution of IPv6
    - Static Routes
    - Applications to test with
  - 2. Add Dynamic Routing to Simple Design Stage
  - 3. Adding Stateless Autoconfiguration via Routers with Prefixes to sections of the network to assign the prefix instead of predefining that prefix
  - 4. Adding a DNS with IPv6 names/addresses
  - 5. Adding DHCP Server capable of updating DNS
  - 6. Test with Tunneling options
- Create documents on “Lessons Learned” for each stage of testing
- Decide on Global IP Addressing Scheme for a Production Network using what has been learned from Testing
- Request IPv6 Global IP Addressing from Regional Internet Registries
- Plan testing stages for production’s topology and new addressing scheme

IBM Advanced Technical Skills (ATS) 38

At each stage of testing you will have opportunities to begin reading the more complex literature on IPv6 implementations.


Smarter Systems for a Smarter Planet
IBM
Avoiding Common IPv6
Implementation Mistakes
-The End-
IBM Advanced Technical Skills (ATS)
39


Smarter Systems for a Smarter Planet
IBM
Avoiding Common IPv6 Implementation Mistakes
-The End-
IBM Advanced Technical Skills (ATS)
40