e-business

Communications Server z/OS V1R6 Technical Update

The journey towards the next generation Internet - IPv6

z/OS V1R6

eserver

Topics

- z/OS V1R6
  - Full sysplex support for IPv6
    - DVIPAs, Sysplex Distributor, SourceVIPA

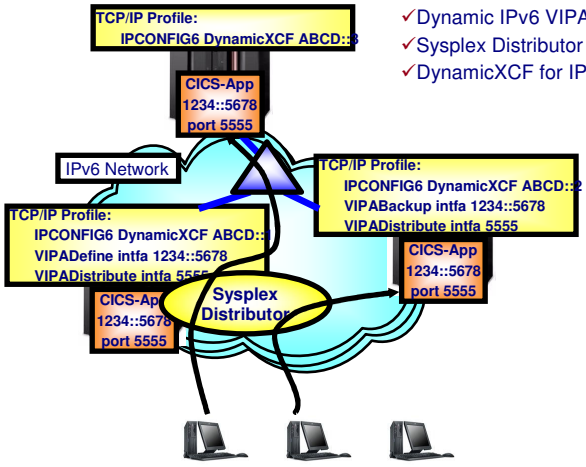
(*) - non-IPv6 enhancements included in this section

e-business

Sysplex support for IPv6

IBM

IPv6 Sysplex support overview



- ✓Dynamic IPv6 VIPA support
- ✓Sysplex Distributor for IPv6 workload
- ✓DynamicXCF for IPv6

IPv6 Sysplex functions in z/OS V1R6

- Dynamic VIPA (DVIPA)
  - VIPA, in general, improves availability since connectivity does not have to be tied to a single physical interface
  - Can be used to identify a network service or application
    - Ex: connect to 1234::5678 vs 2234::2 or 2244::2
- Dynamic VIPA Takeover
  - Improves application availability for planned and unplanned outages
  - Good to setup for backup and recovery
    - Ex: Stack1 is down, stack2 takes over 1234::5678. Stack 2 advertises 1234::5678
- Sysplex Distributor
  - Can improve performance and availability with load balancing between server applications
    - Ex: Sysplex Distributor on Stack 2 load balances connections for CICS on port 5555.
- Underlying connectivity for sysplex uses Dynamic XCF
  - Dynamic XCF supports IPv6 in V1R5
- Other functions associated with sysplex are supported for IPv6 as follows:
  - Sysplex Sockets
  - TCPSTACKSOURCEVIPA
  - Sysplexports
  - Fast Connection Reset after System Failure
  - Enhance Workload Distribution (Application Server Affinity)
  - Dynamically Assign Sysplex Ports
  - Activation of DVIPAs through VIPABACKUP
  - DYNAMICXCF SOURCEVIPAINT
  - Sysplex Distributor Round-Robin Distribution
  - Sysplex Distributor Policy

- The following functions do not support IPv6 in z/OS V1R6:
  - Multinode Load Balancing (MNLB) with CISCO defined with the SERVICEMGR statement
  - Sysplexwide Security Associations (SWSA)
  - HiperSockets
  - MOVEABLE WHENIDLE on the VIPADEFINE statement
  - MOVEABLE DISRUPTIVE on the VIPARANGE statement

Sysplex Functions that Support IPv6

NOTES

- All participating stacks of Dynamic VIPA takeover and Sysplex Distribution of Ports for Load Balancing must be at least V1R6 release
- Sysplex Sockets: This function is via the SO_CLUSTERCONNTYPE option of getsockopt(). Previously, SO_CLUSTERCONNTYPE_NONE was always returned when the SO_CLUSTERCONNTYPE option was used with getsockopt() for an AF_INET6 socket. The proper value will now be returned. The types returned for an IPv6 connection are the same as those returned for an IPv4 connection.
- TCPSTACKSOURCEVIPA: This function allows users to specify a single Dynamic VIPA (DVIPA) or static VIPA to be used as a source IP address for TCP applications that initiate outbound connections on that stack.
- Sysplexports: Sysplex Distributor is enhanced with a facility to allow assignment of ephemeral ports for outbound connections to be managed across the entire sysplex, such that for a particular Distributed DVIPA, a particular port value is assigned to a socket on only one TCP stack in the Sysplex. This will ensure that inbound connection data can always be uniquely routed to the correct application instance.
- Fast Connection Reset after System Failure: This function allows the client stack to notify the client application of a system failure. This improves availability and allows quicker initiation of connection failure recovery. Without this function, the client was unaware of a system failure until it attempted to send data.
  - When target stack fails, the routing stack should attempt to notify the clients of the failure of the connections that terminated in the failed target stack.
  - When stack deletes a DVIPA, the deleting stack should notify any clients of the failed connections.
  - When the routing stack fails, or delete a Distributed DVIPA, and there is no available designated backup stack for that DVIPA, the target stacks should send RST.
- Enhance Workload Distribution (application Server Affinity): This function allows affinities to be established between a specific client (identified by its IP address) and a particular instance of a server application for which work is being balanced with Sysplex Distributor, using a Distributed Dynamic VIPA. This feature ensures that a client that establishes a relationship with a server will be directed to that particular server for subsequent connections. TIMEDAFFinity is an optional parameter on the VIPADISTribute statement that enables this function.

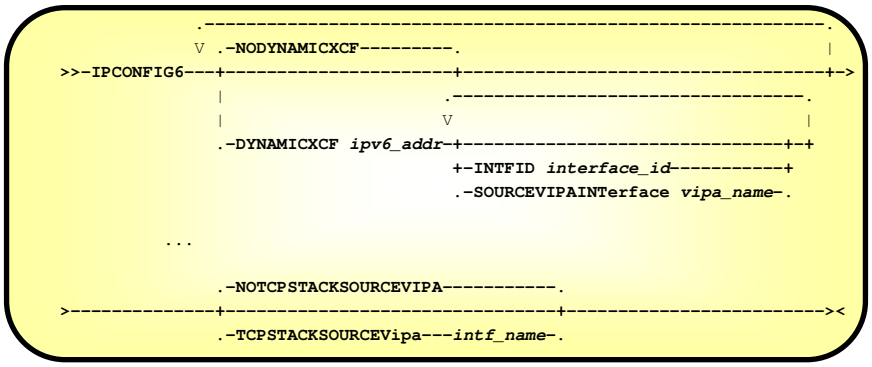
Sysplex Functions that Support IPv6

NOTES

- Dynamically Assign Sysplex Ports: Distributed DVIPAs configured without a PORT parameter on the VIPADISTRIBUTE statement will determine where to distribute work based on where there are applications with listening sockets bound to the distributed DVIPA, regardless of how many different ports are involved. Applications must bind specifically to the designated distributed DVIPA (or have a BIND parameter configured on the PORT statement to accomplish this) in order to be identified as server applications to Sysplex Distributor when no PORT statement is coded on the VIPADISTRIBUTE statement.
- Activation of DVIPAs through VIPABACKUP: VIPABACKUP is enhanced so that an IPv6 Dynamic VIPA may be activated on a backup TCP/IP before it is activated elsewhere in the sysplex with the VIPADEFINE statement. An interface name, a DVIPA IPv6 address and a new MOVEABLE parameter allow this to occur.
- DYNAMICXCF-SOURCEVIPAINT: This function allows the specification of a static VIPA interface to be used as SOURCEVIPA for dynamic XCF link.
- Sysplex Distributor Round-Robin Distribution: Add to the IPv6 VIPADISTRIBUTE statement an optional DISTMethod parameter specifying the method of distributing new connection requests without existing affinity for this Distributed DVIPA. The DISTMethod parameter has two values: BASEWLM (the default distribution that consults WLM and Service Policy Agent for distribution), and ROUNDROBIN (for round-robin distribution to the target stacks).
- Sysplex Distributor Policy: The function is enhanced to support IPv6 addresses for the Policy Agent to Policy Agent connections that are established for the Sysplex Distributor Performance Monitoring function. Policy Agent IPv6 support was already implemented in V1R5.

IPCONFIG6 statement

- Add new optional parameters SOURCEVIPAINTERFACE, and TCPSTACKSOURCEVIPA
- SOURCEVIPA must be enabled for these functions
- For TCPSTACKSOURCEVIPA function, the *intf_name* must be a static VIPA or an active Dynamic VIPA



IPCONFIG6

NOTES

- SOURCEVIPAINTERFACE specifies which static VIPA interface is to be used for SOURCEVIPA. The interface name cannot be a Dynamic VIPA interface. Since the interface may have multiple IP addresses, the source IP address for the outbound connection will be selected based on the default source address selection algorithm.
- For IPv6 TCPSTACKSOURCEVIPA function:
  - The type of addresses that qualify for TCPSTACKSOURCEVIPA will be limited to either a static VIPA or an active Dynamic VIPA (DVIPA).
  - Careful consideration should be given if the interface name is a Dynamic VIPA interface name.
    - The user should specify SYSPLEXPORTS on the VIPADISTRIBUTE statement. Otherwise connections may be disrupted because identical connections could be created from more than one stack.
    - A Dynamic VIPA interface that is created by a VIPARANGE can have multiple dynamic VIPA addresses associated with it. The actual address chosen as the source IP for the outbound connection will not be predictable or necessarily meaningful.
- For existing IPv4 TCPSTACKSOURCEVIPA, there is one minor change: The types of addresses are limited to (same as IPv6)
  - Static VIPAs.
  - Active Dynamic VIPAs.

VIPADEFINE and VIPABACKUP statements

- One IPv6 address per VIPADEFINE statement
- Must specify an interface name
- Allow only IMMEDIATE for MOVEABLE option

```
                    .-MOVEable IMMEDiate-.
>>-VIPADEFine-----+--------------------+--ipv6_intfname--ipv6_addr
```

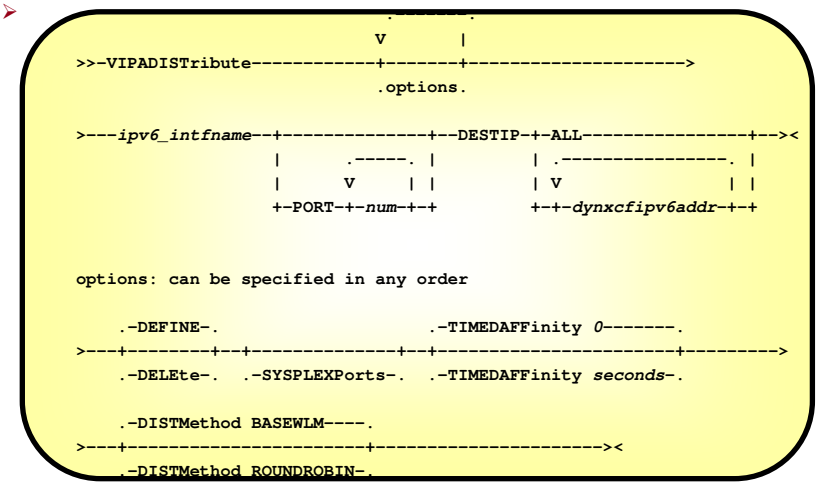
- One IPv6 address per VIPABACKUP statement
- Must specify an interface name
- If MOVEABLE is specified, this DVIPA may be activated on this stack if it is not active elsewhere in the sysplex
- Allow only IMMEDIATE for MOVEABLE option

```
                    .-1----.
>>-VIPABackup-----+------+-->
                    .-rank-.

>-|-ipv6_intfname--ipv6_addr-------------------------------><
    or
  |-MOVEable--IMMEDiate--ipv6_intfname--ipv6_addr----------><
```

VIPADISTRIBUTE statement

- IPv6 DVIPA interface name specified on VIPADISTRIBUTE statement
-



VIPADELETE and VIPARANGE statements

- One IPv6 DVIPA interface name per VIPADELETE statement
- Interface name must be one created by a VIPADEFINE or VIPABACKUP statement.

```
>>-VIPADELete-----ipv6_intfname----------><
```

- The interface name is used for each DVIPA defined from that VIPARANGE statement
- Allow only MOVEable NONDISRUPTive option

```
                    .-DEFINE-.   .-MOVEable NONDISRUPTive-.
>>-VIPARange------+---------+--+-----------------------+---------->
                    .-DELEte-.

>---ipv6_intfname--ipv6_addr/prefix_len---------------------------><
```

Defining Sysplex IPv6

NOTES

- IPv6 specifications cannot be mixed with IPv4 specifications on the same statement, but it is allowed to mix within the VIPADYNAMIC/ENDVIPADYNAMIC block.
- For VIPADEFINE and VIPABACKUP statements you can specify only one IPv6 address for each interface name. You must specify a unique VIPADEFINE or VIPABACKUP statement for each IPv6 interface name and IPv6 address.
- Only one IPv6 interface name can be specified in one VIPADISTRIBUTE statement.
- Only one IPv6 interface name can be specified in one VIPADELETE statement.
- Ensure the interface name for VIPADEFINE/VIPABACKUP is unique within the joined sysplex environment.
- With the VIPARANGE statement, it is possible to define a single interface name and associate all IPv6 DVIPAs in the range with the single IPv6 interface name. OMPROUTE would search for an IPv6_OSPF_Interface statement with the same interface name (or a matching definition using a wildcard).
- DVIPA 1024 limit
  - A stack is limited to no more than 1024 configured or target dynamic VIPAs at any one time. A configured Dynamic VIPA (DVIPA) is one that was created in any of the following ways, and might or might not be active:
    - Using VIPADEFINE
    - Using VIPABACKUP
    - Using an IOCTL SIOCSVIPA or SIOCSVIPA6 DEFINE when this stack has a covering VIPARANGE
    - Using a BIND when this stack has a covering VIPARANGE
    - Using MODDVIPA -c utility
  - The total of 1024 is a combination of all DVIPAs, IPv4 and IPv6.
- Please refer to the description of ipaddr_spec parameter on the INTERFACE Statement for IPAQENET6 for a list of restrictions that must be observed when specifying IPv6 address.

New RACF profile to control who binds to a VIPARANGE DVIPA

- Define a new RACF profile for IPv4 and IPv6 VIPARANGE Bind function
  - There was a security concern for the VIPARANGE NONDISRUPTIVE function. A Dynamic VIPA is created when any application binds to a nonexistent, specific IPaddress falling within a configured VIPARANGE on that stack. The DVIPA ownership is immediately transferred to the second stack without any security or RACF checking when VIPARANGE NONDISRUPTIVE is specified on the first stack. So a new RACF profile (EZB.BINDDVIPARANGE.*sysname.tcpname*) for both IPv4 and IPv6 VIPARANGE function is needed to provide the RACF checking.
  - Define a RACF profile EZB.BINDDVIPARANGE.*sysname.tcpname* in the SERVAUTH class and permit the server started task userIDs. If the profile is not defined, no checking is performed. Non-IBM security products may require the profile to be defined and applications to be permitted.

```
EZB.BINDDVIPARANGE.sysname.tcpname
```

- Planned retirement of functions:
  - Plan to retire MOVEABLE WHENIDLE of IPv4 VIPADEFINE statement in a future release. A warning message is issued when MOVEABLE WHENIDLE is specified.
  - Plan to retire MOVEABLE DISRUPTIVE of IPv4 VIPARANGE statement in a future release. A warning message is issued when DISRUPTIVE is specified.

Sysplex Distributor Policy

- Pagent-to-Pagent communications for Sysplex Distributor performance monitoring
- The Policy Agent on the distributor opens up one or two TCP connections to each of the Policy Agents on the target stacks. The distributor may be configured to distribute connections to IPv4 DVIPAs or IPv6 DVIPAs, or both. The Policy Agent opens an IPv4 connection to a target stack's IPv4 XCF address if it is configured to distribute to an IPv4 DVIPA on that target, and likewise opens an IPv6 connection to a target stack's IPv6 XCF address if it is configured to distribute to an IPv6 DVIPA on that target. As a result, if the routing stack is distributing connections to both IPv4 and IPv6 DVIPAs on a given target, then 2 connections to that target are opened.
- Outboundinterface is used to specify an outbound interface used for Sysplex Distributor distributing stack, it can specify for IPv4 and/or IPv6 addresses. If an address type (IPv4 or IPv6) is not specified, then inbound connections of that address type will be distributed to all available targets.
- An IPv6 address is allowed for the OUTBOUNDINTERFACE attribute on the PolicyAction statement of the policy configuration file

  Outboundinterface *IPv4_addr*

  or

  Outboundinterface *IPv6_addr*

netstat VIPADCFG/-F report

- Netstat VIPADCFG/-F:
-

```
MVS TCP/IP NETSTAT CS V1R6       TCPIP Name: TCPCS           15:47:43
Dynamic VIPA Information:

VIPA Backup:
  IntfName: INTFNAM9
    IpAddr: 2ec0::92b:f209
      Rank: 000100  Moveable:            SrvMgr: n/a
VIPA Define:
  IntfName: INTFNAM1
    IpAddr: 2ec0::92b:f202
      Moveable: Immediate  SrvMgr: n/a
VIPA Range:
  IntfName: INTFNAMSU1
  IpAddr/PrefixLen: 2ec0:888::f202/64
    Moveable: NonDisr
VIPA Distribute:
  DestIntf:  INTFNAM1
  Dest:      2ec0::92b:f202..5011
    DestXCF: 2ec0::943:f003
      SysPt: No   TimAff: 360   Flg: BaseWLM
```

Display stats,type=CFS,strname=EZBEPORT report

- Display stats,type=CFS,strname=EZBEPORT:

```
D NET,STATS,TYPE=CFS,STRNAME=EZBEPORT,SCOPE=ALL,DVIPA=2003::38:1:1
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = STATS,TYPE=CFS 364
IST1370I NETA.SSCP1A IS CONNECTED TO STRUCTURE EZBEPORT
IST1797I STRUCTURE TYPE = LIST
IST1517I LIST HEADERS = 1024 - LOCK HEADERS = 1024
IST1373I STORAGE ELEMENT SIZE = 256
IST924I -------------------------------------------------------------
 .
IST1823I LIST DVIPA SYSNAME  TCPNAME               # ASSIGNED PORTS
IST1824I    1 2003::38:1:1                                      64
IST1825I            VIC038   TCPCS                              64
IST1826I               PORTS:  1024  1025  1026  1027  1028  1029
IST1827I                       1030  1031  1032  1033  1034  1035
IST1827I                       1036  1037  1038  1039  1040  1041
IST1827I                       1042  1043  1044  1045  1046  1047
IST1827I                       1048  1049  1050  1051  1052  1053
IST1827I                       1054  1055  1056  1057  1058  1059
 .
IST1827I                       1078  1079  1080  1081  1082  1083
IST1827I                       1084  1085  1086  1087
IST314I END
```

IPv6 DVIPA SNMP

- SNMP TCP/IP Subagent is enhanced to provide data for IPv6 DVIPA interfaces
  - The SNMP DVIPA management data is defined in the IBM MVS TCP/IP Enterprise-specific MIB module. This MIB module is installed in the HFS directory /usr/lpp/tcpip/samples as file mvstcpip.mi2 and is supported by the TCP/IP Subagent. The existing SNMP DVIPA support has been enhanced as follows:
  - The following existing SNMP MIB tables now support both IPv4 and IPv6 DVIPA interfaces
    - ibmMvsDVIPATable
    - ibmMvsDVIPADistConfTable
    - ibmMvsDVIPAConnRoutingTable
    - ibmMvsDVIPADistPortTable
  - The new ibmMvsDVIPARangeConfigTable supports both IPv4 and IPv6 VIPARANGE values
  - The existing ibmMvsDVIPARangeConfTable will continue to support only IPv4 VIPARANGE values
  - SNMP DVIPA traps will now be created for both IPv4 and IPv6 DVIPA events

Migrating Sysplex applications into an IPv6 environment

- To support IPv6 clients, both TCP/IP stacks and server applications must be modified:
  - TCPIP stack must be dual-mode (both IPv4 & IPv6 interfaces)
    - This allows applications to successfully create AF_INET6 sockets
  - Application binds specifically
    - Explicitly from bind() specific
      - now need two sockets and two binds (one IPv4 and one IPv6)
  - Implicitly via bind on Port reservation in TCPIP profile (typically application binds to inaddr_any/unspecified)
    - Can't specify an IPv4 and IPv6 for same jobname
    - Could start separate instances with separate jobnames if application can support this?
  - bind() to inaddr_any / unspecified (and not modified via PORT reservation BIND option)
    - Probably the most flexible.
- For IPv6 addresses to be used for backup and recovery, partner stacks must be at z/OS V1R6 level or above
- To distribute IPv4 and IPv6 workloads, DYNAMICXCF and DYNAMICXCF6 statements must be defined in the TCPIP profile

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	CICS	IMS	MQSeries	Tivoli
IBM(logo)	Cloudscape	Informix	OS/390	WebSphere
e(logo)business	DB2	iSeries	OS/400	xSeries
AIX	DB2 Universal Database	Lotus	pSeries	zSeries

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds.

Other company, product and service names may be trademarks or service marks of others.

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 and/or changes in the product(s) and/or program(s) 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 (e.g., 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.