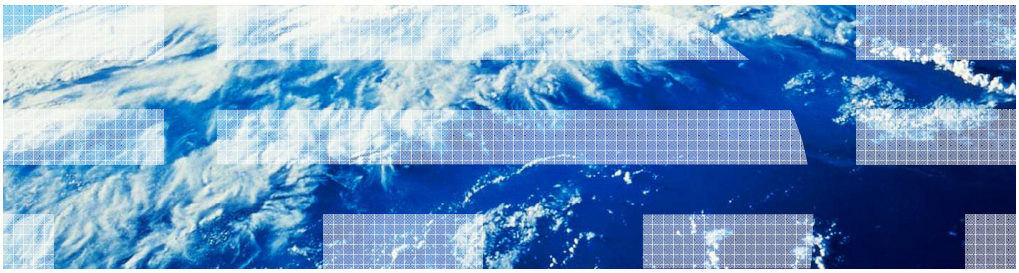
z/OS Communications Server
Application integration, data consolidation, and standards

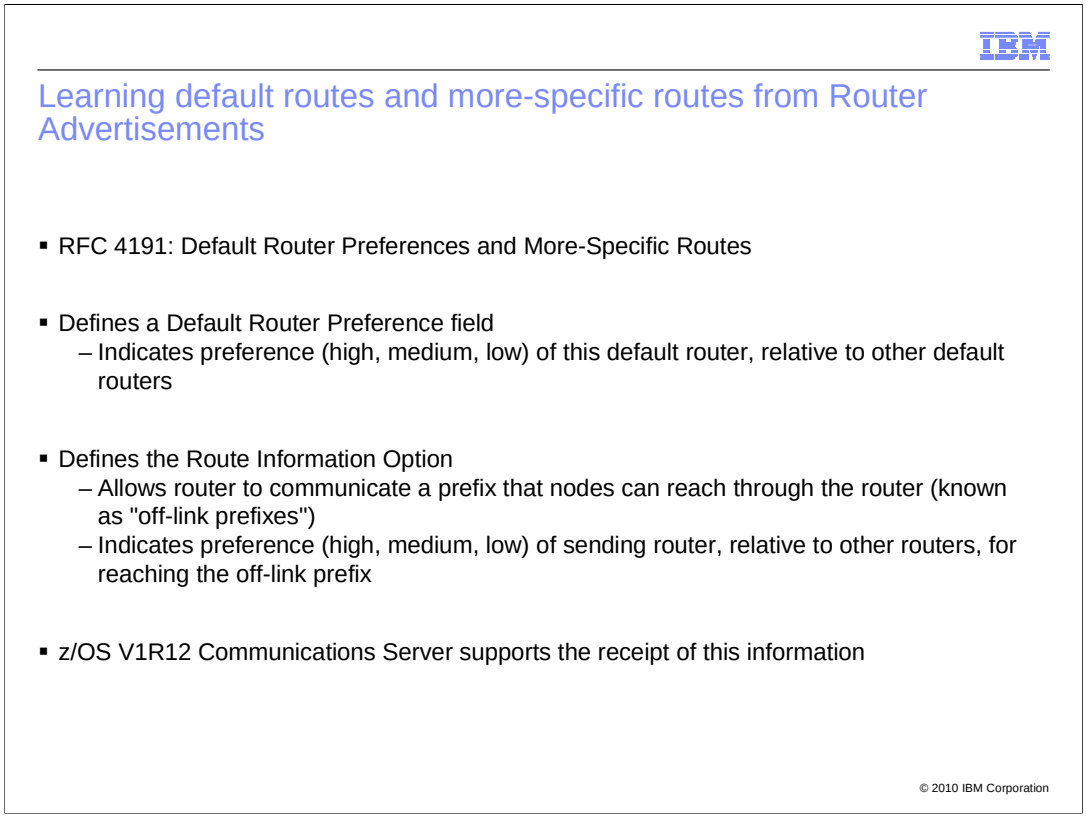


This presentation describes the new functions in z/OS V1R12 Communications Server for application integration, data consolidation, and standards for route and address selection primarily with IPv6.

Enhancements to IPv6 router advertisement include allowing default routers to specify a preference field to identify which one or ones should be preferred for default routes. Routers can also specify off-link prefixes which can be reached through the router and, like default routes, can include a preference weight.

You now have the ability to configure the default address selection policy table that is used for both source address selection and destination address selection.

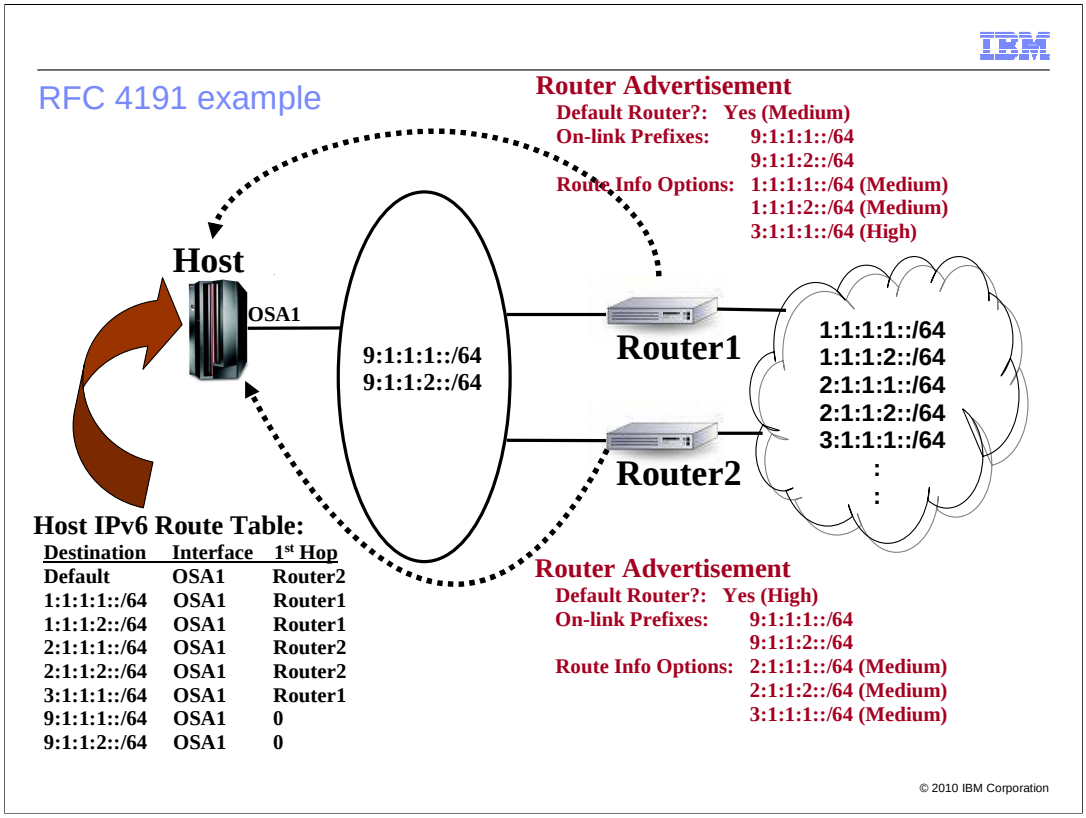
New socket APIs are provided that allow applications to specify whether applications prefer a temporary or public IPv6 source address.



In some network topologies where a node has multiple routers on its Default Router List, the choice of which router is used to reach an off-link destination is important. For example, in some situations, one router might provide much better performance than another for a destination. In other situations, choosing the wrong router might result in a failure to reach the destination. Although the choice might be important, the router advertisement message as originally defined does not provide a way to control which router is chosen.

RFC 4191 solves the problem by defining a new field, called the Default Router Preference, for the router advertisement message. The sending router uses this field to indicate whether it should be preferred over other routers as a default router. RFC 4191 also defines a new option, called the Route Information Option, for the router advertisement message. This option allows a router to communicate a prefix that the nodes can reach through the router. These prefixes are known as off-link prefixes. The Route Information Option includes a field, called the Route Preference, that indicates whether the sending router should be preferred over other routers for reaching the off-link prefix.

These changes to the router advertisement message improve a node’s ability to pick an appropriate router for an off-link destination. z/OS V1R12 Communications Server supports the receipt of this new field and new option.



This example illustrates the new RFC 4191 processing. The routers, identified as Router1 and Router2, each send a router advertisement message to the nodes on the link. Both router advertisement messages indicate the same two prefixes as being present on the link. They both also indicate that the node should use the sending router as a default router. Router1 indicates a default router preference of medium, while Router2 indicates a default router preference of high.

Router1 uses Route Information Options to indicate that the node can use Router1 to reach three off-link prefixes. Router2 uses Route Information Options to indicate that the node can use Router2 to reach three off-link prefixes. One of the prefixes is included in Route Information Options by both routers. For that prefix, Router1 indicates a preference of high, while Router2 indicates a preference of medium.

As a result, the IPv6 route table on the node contains a default route with Router2 as first hop, due to Router2 indicating a default router preference of high. It contains prefix routes with Router1 as first hop for the two prefixes only included in Route Information Options by Router1. It contains prefix routes with Router2 as first hop for the two prefixes only included in Route Information Options by Router2. It contains a prefix route with Router1 as first hop for the prefix included in a Route Information Option by both Router1 and Router2. This is because Router1 indicated a preference of high, while Router2 indicated a preference of medium. Finally, it contains direct routes to the two on-link prefixes.

Precedence within routes

1. Non-replaceable static routes
2. OSPF routes
3. RIP routes
4. **Router advertisement routes** ← choice now required **within** this type
5. Replaceable static routes

- Which route is installed in the route table when router advertisement routes to the same destination are received from multiple routers?
  1. 'Reachable' routes take precedence over 'unreachable' routes
  2. Highest preference reachable or unreachable routes are installed
- An exception to the route precedence rules is that unreachable router advertisement routes are lower precedence than any replaceable static routes
- If unreachable router advertisement routes are installed, they are flagged as inactive

The same five route precedence rules are in effect for z/OS V1R12 with an enhancement to rule four. The highest precedence is for non-replaceable static routes. The next highest precedence is for OSPF routes followed by RIP routes. The fourth highest precedence is router advertisement routes. The lowest precedence is replaceable static routes.

With z/OS V1R12, it is now possible for the TCP/IP stack to learn router advertisement routes, for the same destination, with different preference values. In addition, the routers that have originated router advertisement messages can become unreachable, making the routes through them unusable. An originating router can become unreachable when the neighbor discovery protocol detects that the router is no longer responding or when the interface to the router becomes inactive. The stack uses precedence rules to determine the routes to install in the route table when there is a combination of reachable and unreachable routes and routes with different preferences. (A reachable router is a reachable neighbor (according to Neighbor Discovery) and the interface is active.)

First, if the router advertisement routes for a destination are a combination of reachable and unreachable routes, the candidate routes for installation are all of the reachable routes. Otherwise, the candidate routes are all available router advertisement routes. Next, the routes to install are selected from the candidate routes according to route preference. Only the candidate routes at the highest preference are installed.

One exception to the precedence rules occurs when all of the router advertisement routes to a destination are unreachable and there are replaceable static routes configured for that destination. In this case, the replaceable static routes are installed.

When unreachable router advertisement routes are installed in the route table, they are flagged as inactive routes. This means that the routes are not used if there is a less specific route for the destination that can be used as an alternative. In addition, Netstat route reports will display these unreachable routes without the Up flag (U). This is the same behavior as is used for static routes that are inactive.

Function externals: Netstat metric

Netstat ROUTE/-r DETAIL

```
MVS TCP/IP NETSTAT CS V1R12       TCPIP Name: TCPCS3          14:43:03
IPv6 Destinations
DestIP:   Default
  Gw:     fe80::7
  Intf:   NSQDIO3L6         Refcnt:  0000000000
  Flgs:   UGD               MTU:     9000
  Metric: 00000002
  MVS Specific Configured Parameters:
    MaxReTransmitTime:  120.000   MinReTransmitTime: 0.500
    RoundTripGain:      0.125     VarianceGain:      0.250
    VarianceMultiplier: 2.000     DelayAcks:         Yes
```

- Metric for router advertisement routes reflects advertised preference
  - 0 = direct route (on-link prefix) – no associated preference
  - 1 = indirect route (default or off-link prefix) with high preference
  - 2 = indirect route (default or off-link prefix) with medium preference
  - 3 = indirect route (default or off-link prefix) with low preference

The metric value displayed on Netstat route reports for router advertisement routes now indicates the preference value advertised for the route. If the route is a direct prefix route, for which no preference value can be advertised, a metric of zero is displayed. For all other router advertisement routes, a metric of one indicates high preference, a metric of two indicates medium preference, and a metric of three indicates low preference.

Function externals: OMPROUTE metric

- RT6TABLE

```
EZZ7979I IPV6 ROUTING TABLE 636
DESTINATION: ::/0
  NEXT HOP: FE80::7
  TYPE: RADV           COST:  2         AGE: 6
```

- Metric (cost) for router advertisement routes reflects advertised preference
  - 0 = direct route (on-link prefix) – no associated preference
  - 1 = indirect route (default or off-link prefix) with high preference
  - 2 = indirect route (default or off-link prefix) with medium preference
  - 3 = indirect route (default or off-link prefix) with low preference

The metric (cost) value displayed on OMPROUTE route reports for router advertisement routes will also now indicate the preference value advertised for the route. The various metric values will indicate the same preference values as they do on the Netstat reports.

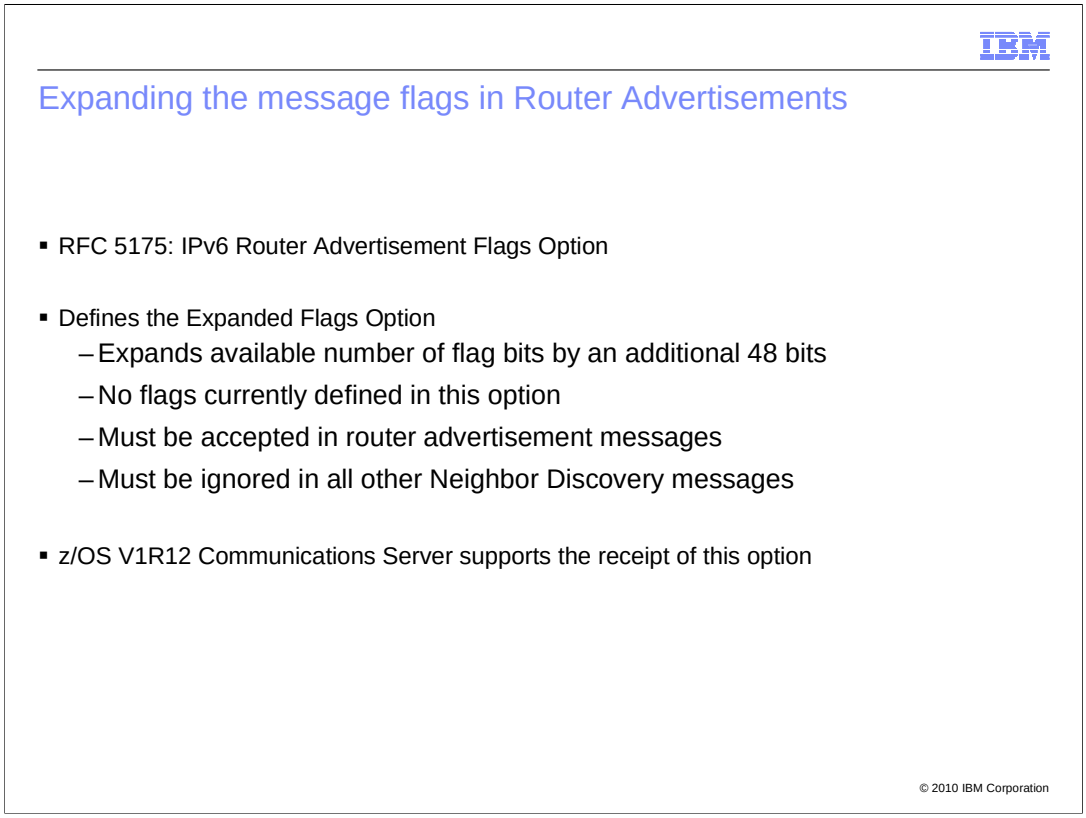
Netstat display of uninstalled routes

- New modifier on Netstat ROUTe/-r command: RADV
- Displays all received router advertisement routes
- Can be used together with DETAIL modifier
  - Displays route information that was previously not displayable

Router advertisement routes that have been received but are not installed, due to the existence of higher precedence routes, cannot be displayed. The inability to display router advertisement routes that are not installed due to the existence of higher precedence routes is solved by a new modifier on the Netstat ROUTE command. When this new modifier is specified, all received router advertisement routes are displayed regardless of whether they are installed in the route table. The new modifier can be used together with the DETAIL modifier to display detailed information about the router advertisement routes. This combination of modifiers allows for the display of information that was previously not able to be displayed.

A new modifier has been added to the Netstat ROUTE command. The new modifier displays all router advertisement routes that have been received, regardless of whether they are currently installed. All types of router advertisement routes are included in the report, default routes, on-link prefix routes, and off-link prefix routes.

When the new modifier is used together with the DETAIL modifier, detailed information is displayed for the router advertisement routes. There are four fields of particular interest for router advertisement routes. The metric field indicates the preference value that was received for the route. The lifetime expiration field indicates, in Greenwich Mean Time, when the route will expire and be deleted if it is not refreshed by a new router advertisement from the router. The gateway reachable field indicates whether the Neighbor Discovery protocol has discovered that the router is reachable. This field can indicate a reason why a route might be uninstalled or inactive. Finally, the interface active field indicates whether the interface used by the route is active. This field can also indicate a reason why a route might be uninstalled or inactive.

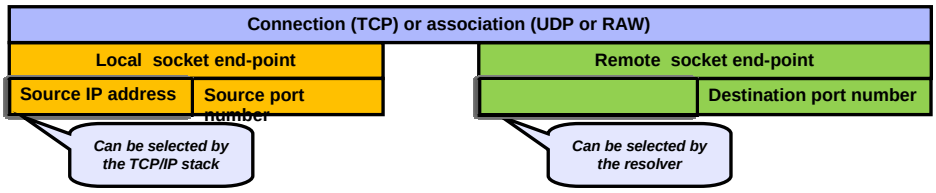


The field reserved for flags in the router advertisement message is only an eight bit field. Several protocols have reserved flags in this field and others are preparing to reserve a significant number of flags such that the field will soon be exhausted.

The flag shortage problem is solved by RFC 5175, IPv6 Router Advertisement Flags Option. RFC 5175 solves the problem by defining a new option, called the Expanded Flags Option, for the router advertisement message. This option expands the available number of flag bits by an additional 48 bits, although none of them are yet defined for use. The Expanded Flags Option must be accepted in received router advertisement messages and must be ignored in all other messages of the Neighbor Discovery protocol.

z/OS V1R12 Communications Server supports the receipt of this new option.

Source and destination IP address selection overview



- Sockets programs can specify all four elements entirely, but do not have to
- The TCP/IP stack can choose
  - Source IP address
  - Source port number
- The resolver can choose
  - Destination IP address
  - Destination port number
- RFC 3484 *"Default Address Selection for Internet Protocol version 6 (IPv6)"*
  - Defines configurable rules for the source and destination IP address selection logic
  - This logic kicks in after all the existing z/OS TCP/IP logic for selection of source and destination IP addresses has been exhausted

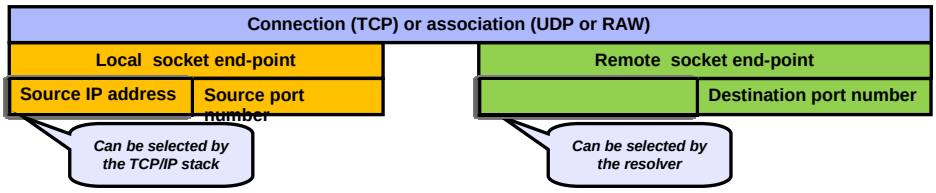
For sockets programs, five elements are used to identify a given socket connection. The five elements are the protocol, such as TCP; the source IP address; the source port number; the destination IP address; and the destination port number. A sockets program can specify some or all of these elements, or might let the TCP/IP stack and resolver do so for the application.

The TCP/IP stack can select the source IP address for an application that does not explicitly specify bind to a source IP address. z/OS TCP/IP provides many ways to influence this selection through the source VIPA functions. The TCP/IP stack can also select the source port number from the available pool of ephemeral port numbers.

The resolver can select the destination IP address, using the default address selection algorithm defined in RFC 3484. The resolver can also select the destination port number if the application uses the getservbyname function.

RFC 3484 defines a set of rules that are used in performing both source IP address selection and destination IP address selection. While the RFC provides a "default" set of rules to be used, it also suggests that products allow the rules to be customized by the user.

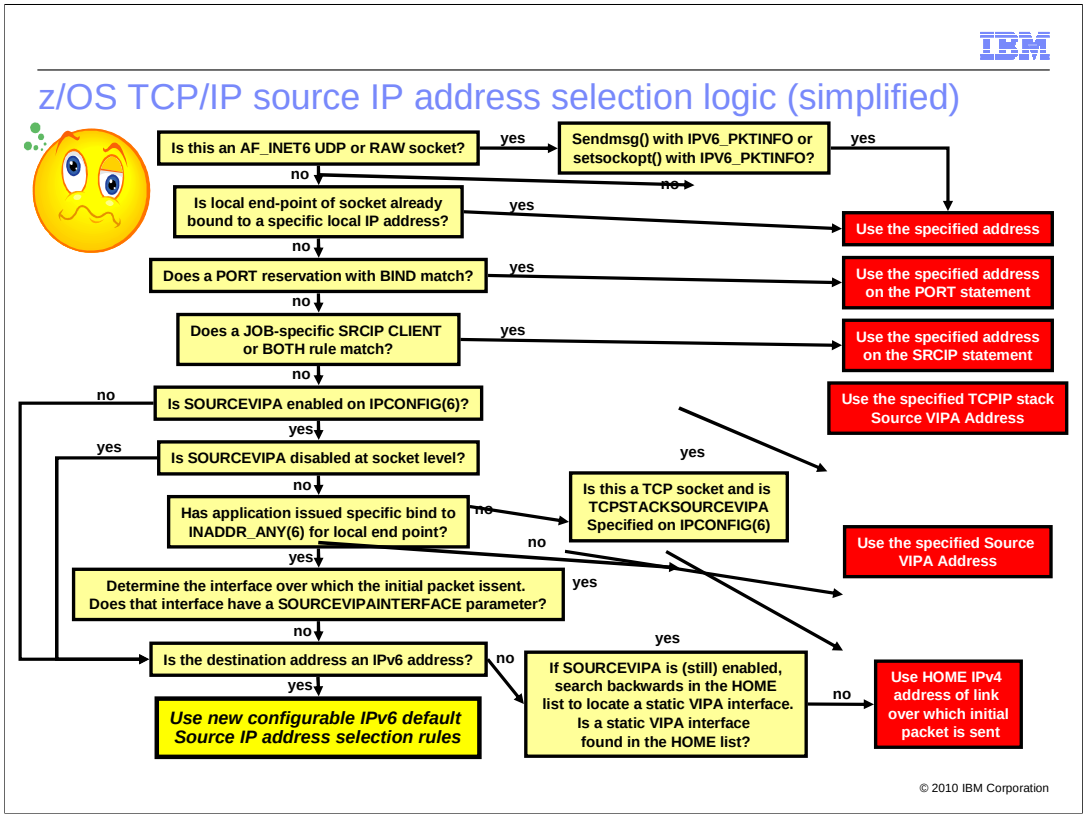
RFC 3484 implications – primarily implications for IPv6



- **Source address selection**
  - No impact when destination is an IPv4 address
  - For IPv6 destinations, the new configurable rules kick in if neither SOURCEVIPA nor SRCIP selects a source IP address
  - New rules configurable by way of a new TCP/IP profile statements
- **Destination address selection**
  - Governs the order in which IP addresses are returned by the getaddrinfo() resolver call
  - No changes for gethostbyname()
  - No changes if IPv6 is not enabled
  - SORTLIST continues to govern order of IPv4 addresses
  - New configurable rules can be used to alter preference for IPv6 over IPv4 addresses to the opposite, but otherwise no impact to IPv4 destinations

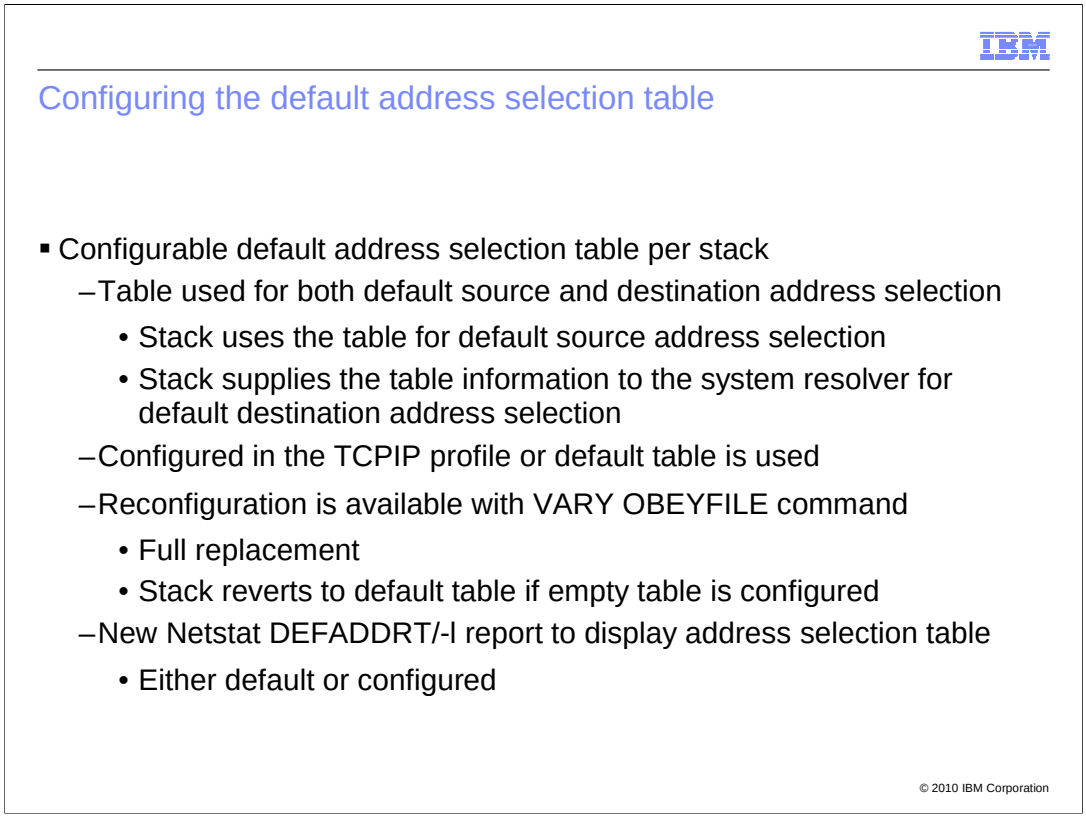
The default address selection algorithms defined in RFC 3848 are not always applied. For the source IP address selection, the algorithms only apply to IPv6 destinations. Even then, the algorithms are only used if the application does not bind to an IP address, and neither SOURCEVIPA, SRCIP nor the BIND parameter on the PORT statement selects a source IP address.

For destination IP address selection, the algorithms are only applied if IPv6 is enabled for the TCP/IP stack. It also requires the application to be enabled for IPv6 and use the getaddrinfo function. No changes are made if the application uses the older gethostbyname function. Fortunately, the vast majority of applications are already using getaddrinfo.



z/OS Communications Server provides many ways to influence the source IP address that is used for a new connection. The application itself can select the source IP address in two ways. The application can bind to a specific local IP address or, in the case of UDP and RAW sockets, use the IPV6_PKTINFO option on either the sendmsg or the setsockopt functions. You can use the BIND parameter on the PORT reservation statement, or use the SRCIP statement, to select the source IP address. SOURCEVIPA, TCPSTACKSRCVIPA, or SOURCEVIPAINTERFACE can be used to select a source VIPA address.

Only if none of these selects a source IP address, and the destination is an IPv6 address, then the TCP/IP stack applies the default source address selection algorithms as defined in RFC 3484.



z/OS V1R12 Communications Server implements the recommendations of RFC 3484 and allows the system administrator to configure a default address selection table for a TCP/IP stack. A single table is used for both source and destination address selection. The table is configured in the TCP/IP profile and, if not configured, the stack will use the default selection table as defined in RFC 3484. Dynamic reconfiguration of the table is available with the VARY OBEYFILE command. VARY OBEYFILE will cause the default address selection table to be completely replaced or deleted. The TCP/IP stack supplies the table information to the system resolver which then uses the information for default destination address selection. A new **Netstat DEFADDRT/-l** report will show either the default (non-configured) table or the configured table.

Configuring the default address selection table: default table

```
; Prefix          Precedence Label
DEFADDRTABLE
      ::1/128         50         0
      ::/0            40         1
      2002::/16       30         2
      ::/96           20         3
      ::ffff:0:0/96   10         4
ENDDEFADDRTABLE
```

- Note: The table shown here matches the default table

This is an example default address selection table. The table as shown is identical to the table that is used if one is not configured.

The first entry in the table (::1/128) is for IPv6 loopback. The second entry (::/0) is for native IPv6 addresses and will match any IPv6 address that does not match one of the other rows in the table. The third entry (2002::/16) is for 6-to-4 tunnel addresses. The fourth entry (::/96) is for IPv4-compatible addresses. The last entry (::ffff:0:0/96) is for native IPv4 addresses.

The address selection algorithms prefer addresses with matching labels. This table prefers using native source addresses with native destination addresses, 6to4 source addresses with 6to4 destination addresses, and IPv4-compatible source addresses with IPv4-compatible destination addresses.

This destination address selection algorithm prefers addresses with higher precedence. This table prefers IPv6 destination addresses over IPv4 destination addresses since the entry for IPv4-mapped addresses (::ffff:0:0/96) has the lowest precedence.

By changing the precedence for the IPv4-mapped address from 10 to 100, the table will prefer IPv4 destination addresses over IPv6 destination addresses since the entry for IPv4-mapped addresses, ::ffff:0:0/96, has the highest precedence.

Netstat display of the default address table (TSO)

```
NETSTAT DEFADDRT

MVS TCP/IP NETSTAT CS V1R12       TCPIP Name: TCPCS   20:30:49
Policy Table for IPv6 Default Address Selection:
  Source: Default
Precedence Label Prefix
---------- ----- ------
50         0     ::1/128
40         1     ::/0
30         2     2002::/16
20         3     ::/96
10         4     ::ffff:0:0/96
```


This is an example of the output from the **NETSTAT DEFADDRT** command in the TSO environment. The output shows a title line identifying the Netstat release and TCP/IP stack name followed by a title line identifying the report as a policy table report. This is followed by a line indicating if the policy table that is displayed is configured or if it is the default table and, therefore, not configured. This is then followed by a header with three columns identified as Precedence, label and prefix. This is followed by the configured or default values for the table.

Socket API extensions to prefer temporary IPv6 addresses – RFC 5014

- RFC-5014 defines:
  - socket API extensions to implement RFC-3484 source address selection rule 7 (temporary versus public address preference)
  - additional source IP address preferences for applications
  - getaddrinfo() extensions to allow system resolver to order returned destination IP addresses based on application-specified source IP address preferences and configured address selection table
  - two new functions to assist in binding a socket to a preferred source address and verifying address properties
- Implementation of RFC-5014 increases standards compliance and application portability

RFC 5014 defines API extensions to enable an IPv6 socket application to prefer a temporary IPv6 address over a public IPv6 address. RFC 5014 defines socket options that enable an application to specify source IP address selection preferences. It also defines new flags to pass to getaddrinfo() to modify IP address selection based on application source IP address selection preferences.

RFC 5014 defines two new API functions to assist in binding to a preferred source address and verifying an address against source address preferences.

RFC 5014 defines socket API extensions that increase standards compliance and application portability.

Socket API extensions to prefer temporary IPv6 addresses

- The following functions are new or extended for RFC-5014 support
  - New bind2addrsel() socket function
  - New inet6_is_srcaddr() function
  - New IPv6 socket option for setsockopt() and getsockopt() functions
  - New parameter for getaddrinfo() function
  - New return message from gai_strerror() function
- The new functions apply to both stream (TCP) and datagram (UDP) IPv6 sockets
- RAW sockets are not supported

RFC 5014 defines two new functions: bind2addrsel() is used for a new type of bind and inet6_is_srcaddr() is a function used to query an IPv6 address to see if it meets application-specified requirements. Additionally, there is a new IPv6 socket option to express source address selection preferences for setsockopt() and getsockopt(). The addrinfo structure used by the getaddrinfo() function is extended to allow passing in additional flags to specify source address selection preferences.

A new (additional) return code from getaddrinfo() is defined and gai_strerror() has been updated to translate this new return code to printable form.

The new functions and options apply to both stream (TCP) and datagram (UDP) AF_INET6 sockets only; Raw sockets are not supported.

New socket option and flags for IPv6 (AF_INET6) sockets

Flag name	Meaning
IPV6_PREFER_SRC_HOME	Prefer Home address as source
IPV6_PREFER_SRC_COA	Prefer Care-of address as source
IPV6_PREFER_SRC_TMP	Prefer Temporary address as source
IPV6_PREFER_SRC_PUBLIC	Prefer Public address as source
IPV6_PREFER_SRC_CGA	Prefer Cryptographically Generated address (CGA) as source
IPV6_PREFER_SRC_NONCGA	Prefer a non-CGA address as source

unsigned int flags;
setsockopt(s, IPPROTO_IPV6, IPV6_ADDR_PREFERENCES, flags, …);
getsockopt(s, IPPROTO_IPV6, IPV6_ADDR_PREFERENCES, &flags, …);

RFC 5014 defines these flags which applications can use to alter the default source address selection rules discussed earlier in this presentation.

The new socket options flags for IPPROTO_IPV6, IPV6_ADDR_PREFERENCES can be used singularly, or in combination, to specify the application's preference. RFC 5014 defines how to handle contradictory combinations in addition to flags that correspond to types of IP addresses that are not implemented.

z/OS Communications Server does not implement cryptographically generated addresses (CGA) or Care-of addresses (COA). Applications can use the defined flags to express a preference for either of these (or both) but the preference cannot be satisfied. These flags are passed to setsockopt() and returned by getsockopt(). Default values returned by getsockopt() if no preferences have been set are: IPV6_PREFER_SRC_HOME, IPV6_PREFER_SRC_PUBLIC and IPV6_PREFER_SRC_NONCGA.

The flags are called preferences because there is no guarantee that the application's preferences are satisfied. A new function called inet6_is_srcaddr can be used to test addresses against a set of flags.

The named constants shown here are the RFC-defined constants used for the C API. Non-C APIs use similar names.

V1R11: prefer temporary addresses using the SRCIP statement

- z/OS Communications Server currently supports the TEMPADDRS keyword on the SRCIP statement
  - Ability to change rule7 of source address selection to prefer temporary IPv6 addresses over public IPv6 addresses when all preceding rules result in a tie
- Temporary IPv6 addresses are created when the TCPIP profile statement, IPCONFIG6 TEMPADDRS, is used and IPv6 routers request them
- Temporary addresses, while not a security mechanism, provide better privacy for clients

Support for temporary IPv6 addresses was added in z/OS V1R11 Communications Server.

The TEMPADDRS keyword on the SRCIP profile statement enables the TCP/IP stack to reverse Rule7 of default source address selection and prefer temporary addresses over public addresses.

Temporary addresses are created when IPCONFIG6 TEMPADDRS has been specified in the TCP/IP profile and a router advertisement is received requesting temporary addresses.

Temporary addresses provide better privacy for clients since the address is not fixed and is not easily correlated to a specific client.

V1R12: Prefer public addresses using the SRCIP statement

- RFC-5014 introduced an API to allow an application to specify a preference for public or temporary IPv6 addresses
  - If the application specifies a preference for temporary IPv6 addresses, the administrator will need a way to override the application to use public addresses
  - The current SRCIP statement supports the TEMPADDRS keyword but does not have a way to specify an override preference for public addresses
- A new PUBLICADDRS parameter on the SRCIP profile statement to specify a preference for public IPv6 addresses
  - SRCIP statement already has a TEMPADDRS parameter to specify a preference for temporary IPv6 addresses
  - This supersedes any preference specified by the application
  - Applies to all protocols (TCP,UDP,RAW)
- Existing **Netstat SRCIP/-J** report updated to show new option

The API allows an application to specify a preference for public or temporary addresses. This creates a problem for the system administrator. If you want to require temporary addresses, but the application specifies a preference for public addresses, you can use the SRCIP statement in the TCP/IP profile to override the application and specify a preference for temporary addresses.

The opposite is not true. If the system administrator wants to require public addresses, but the application specifies a preference for temporary addresses, there is no way to override the application and specify a preference for public addresses.

A new parameter, PUBLICADDRS, on the SRCIP statement allows the system administrator to specify a preference for public addresses. This new parameter is in addition to the current parameter, TEMPADDRS, which specifies a preference for temporary addresses. This option will override any preference specified by the application.

The **Netstat SRCIP/-J** report is updated to show the new option.

NETSTAT SRCIP example

```
MVS TCP/IP NETSTAT CS V1R12       TCPIP Name: TCPCS          20:30:49
Source IP Address Based on Job Name:
Job Name  Type  Flg  Source
--------  ----  ---  ------
*         IPV4  C    9.67.5.16
*         IPV6  C    DVIPA66
T*        IPV4  S    9.67.5.15
T*        IPV6  S    2000::9:67:5:15
TCPUSR1*  IPV4  B    9.67.5.12
TCPUSR2*  IPV6  B    DVIPA62
TCPUSR3*  IPV6       TEMPADDRS
TCPUSR4*  IPV6       PUBLICADDRS
U*        IPV4  C    9.67.5.14
U*        IPV6  C    DVIPA64
USER*     IPV6  C    2000::9:67:5:13
USER1*    IPV4  C    9.67.5.13
USER12    IPV4  C    9.67.5.11
U27       IPV6  C    2000::9:67:5:11

Source IP Address Based on Destination:
Destination: 10.1.0.0/16
  Source:    9.1.1.2
Destination: 10.1.1.1
  Source:    9.1.1.1
Destination: 2001:0db8::0522:f103
  Source:    2000::9:67:5:10
Destination: 2001:0db8::/32
  Source:    DVIPA66
```


This example shows the output from a **NETSTAT SRCIP** report. The report is not new for z/OS V1R12 Communications Server but a new value of “PUBLICADDRS” can now appear in the source column.

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 email feedback:

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

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

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

Trademarks, disclaimer, and copyright information