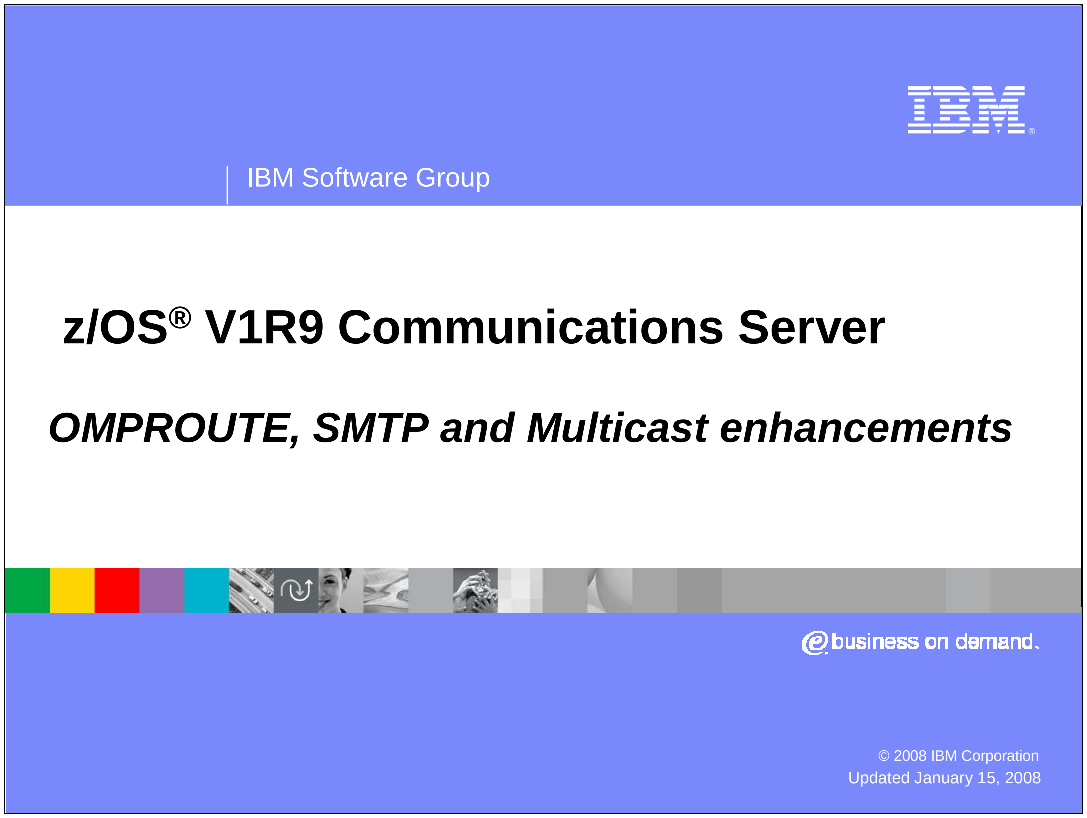


IBM Software Group

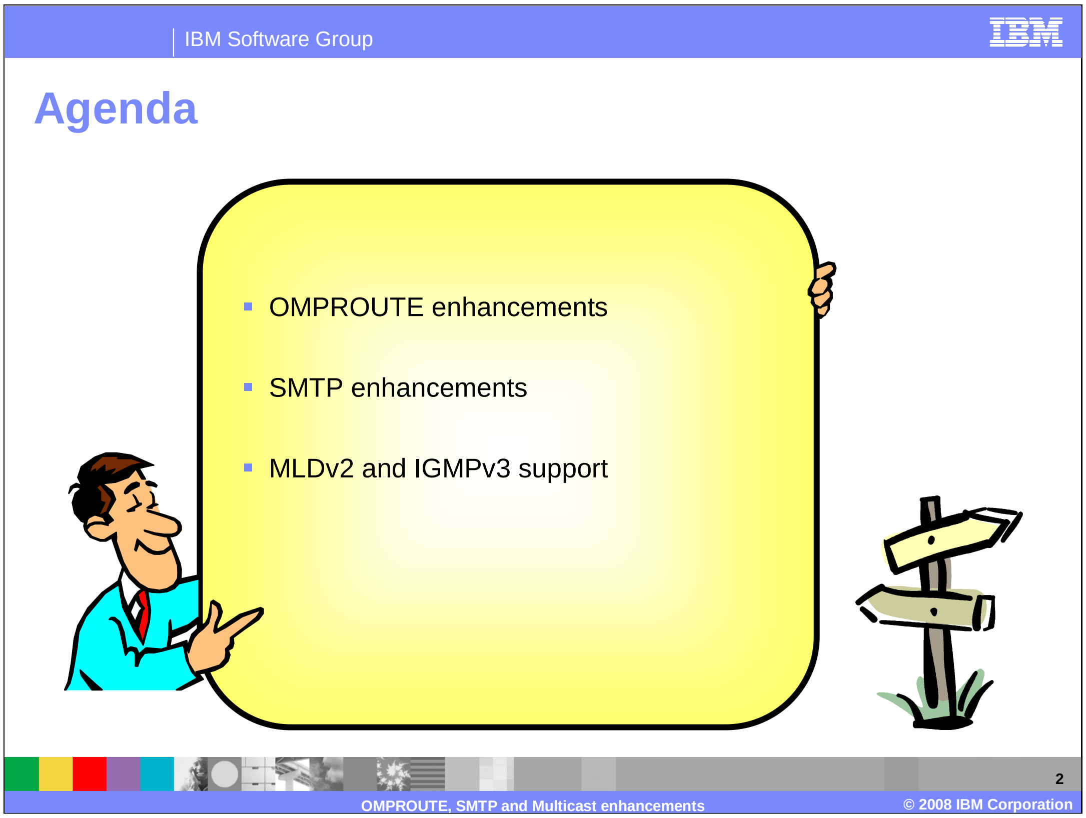
z/OS® V1R9 Communications Server

OMPROUTE, SMTP and Multicast enhancements

business on demand.

Updated January 15, 2008

This presentation describes OMPROUTE, SMTP and Multicast enhancements to z/OS V1R9 Communications Server.



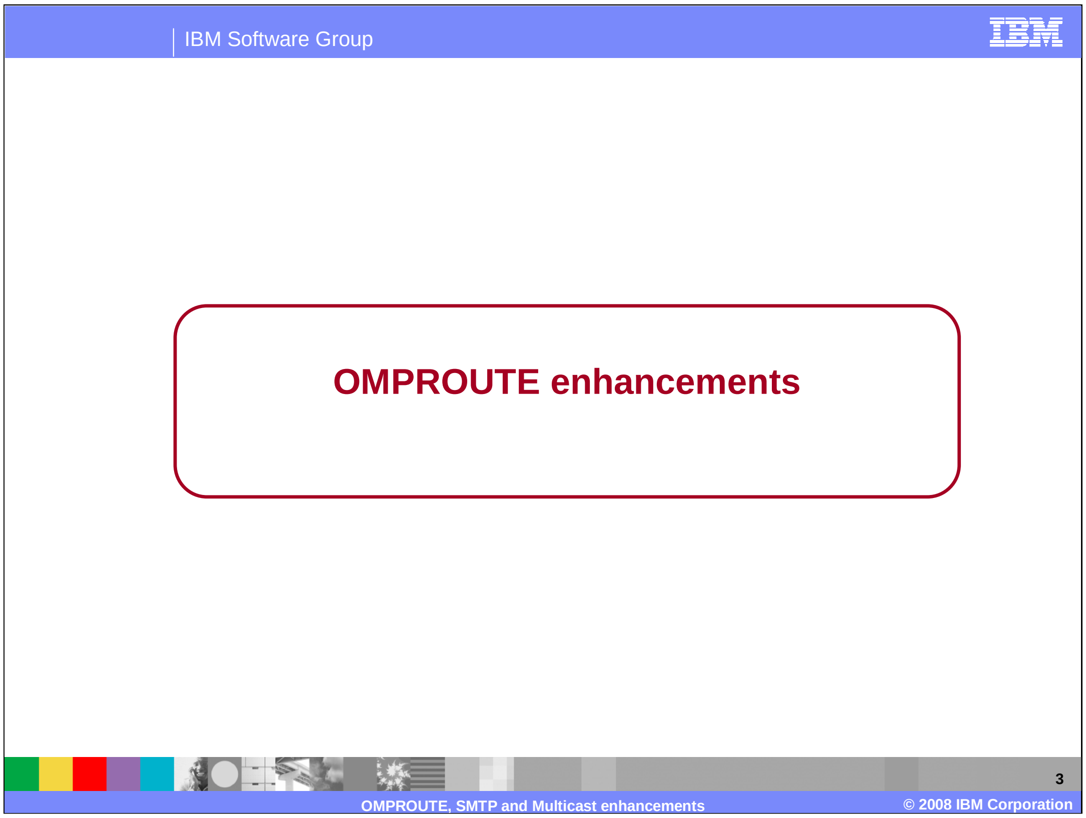
Agenda

- OMPROUTE enhancements
- SMTP enhancements
- MLDv2 and IGMPv3 support

There are several OMPROUTE enhancements in V1R9. These enhancements stem from requirements that were made.

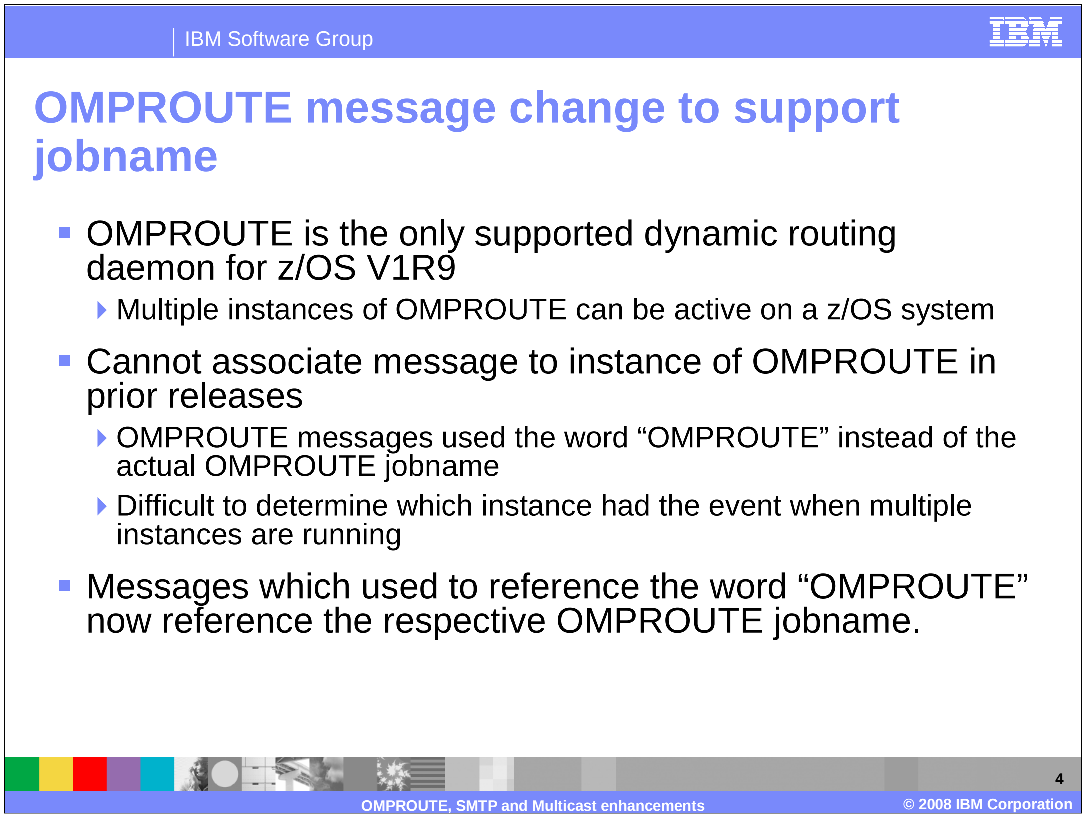
SMTP Enhancements consist of a new SMTP configuration statement (REMOTEPORT) and new operator commands.

The new versions of the Internet Group Management Protocol for IPv4 , IGMPv3, and Multicast Listening Discovery for IPv6, MLDv2 were implemented in z/OS V1R9 Communications Server.



OMPROUTE enhancements

This section covers the enhancements made to OMPROUTE in z/OS V1R9 Communications Server.



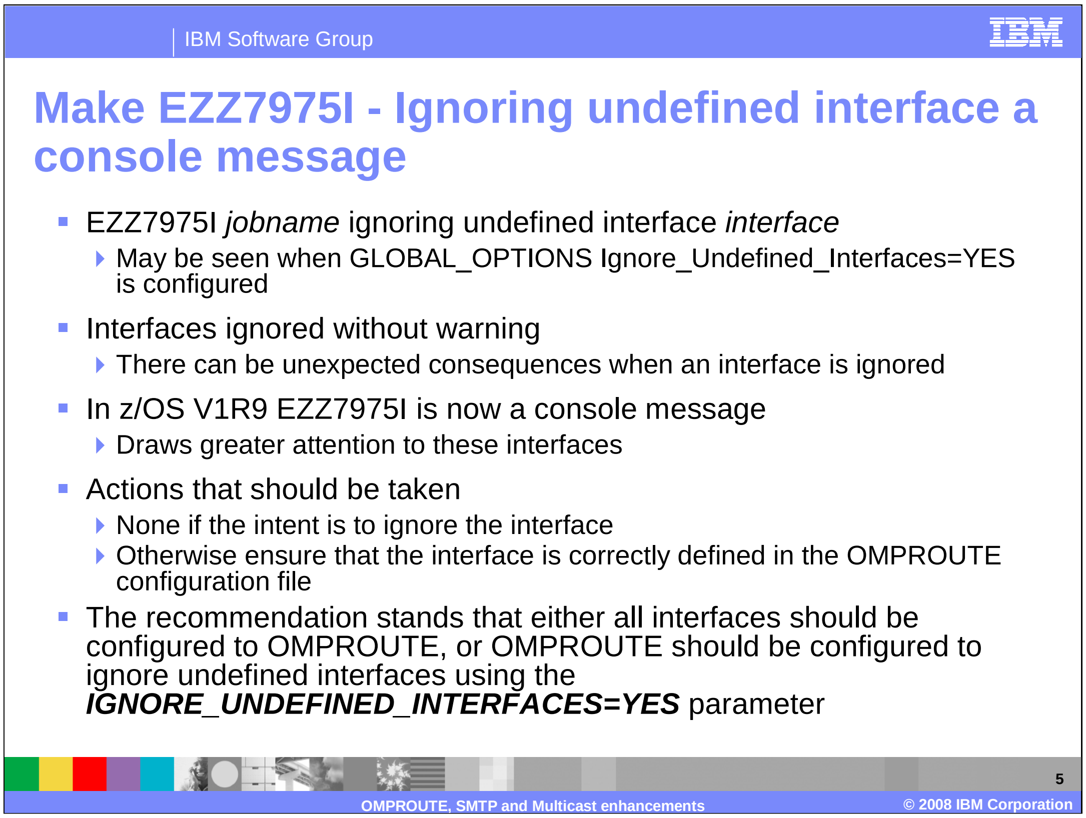
OMPROUTE message change to support jobname

- OMPROUTE is the only supported dynamic routing daemon for z/OS V1R9
  - Multiple instances of OMPROUTE can be active on a z/OS system
- Cannot associate message to instance of OMPROUTE in prior releases
  - OMPROUTE messages used the word “OMPROUTE” instead of the actual OMPROUTE jobname
  - Difficult to determine which instance had the event when multiple instances are running
- Messages which used to reference the word “OMPROUTE” now reference the respective OMPROUTE jobname.

OMPROUTE is the only supported dynamic routing daemon in z/OS V1R9. It supports RIP, OSPF, IPv6 RIP and IPv6 OSPF routing protocols. In a Common INET environment a copy of OMPROUTE must be started for each stack that requires OMPROUTE services. Therefore multiple instances of OMPROUTE can be active on a z/OS system.

Prior to z/OS V1R9, some OMPROUTE messages used the word “OMPROUTE” instead of the actual OMPROUTE jobname to indicate which instance of OMPROUTE triggered the message. If more than one OMPROUTE instance is running, and a message indicated an event by “OMPROUTE”, there was potential for error to determine which instance of OMPROUTE was associated with the event.

Messages which previously referenced “OMPROUTE”, now reference the jobname for the instance of OMPROUTE which triggered the message.



Make EZZ7975I - Ignoring undefined interface a console message

- EZZ7975I *jobname* ignoring undefined interface *interface*
  - May be seen when GLOBAL_OPTIONS Ignore_Undefined_Interfaces=YES is configured
- Interfaces ignored without warning
  - There can be unexpected consequences when an interface is ignored
- In z/OS V1R9 EZZ7975I is now a console message
  - Draws greater attention to these interfaces
- Actions that should be taken
  - None if the intent is to ignore the interface
  - Otherwise ensure that the interface is correctly defined in the OMPROUTE configuration file
- The recommendation stands that either all interfaces should be configured to OMPROUTE, or OMPROUTE should be configured to ignore undefined interfaces using the ***IGNORE_UNDEFINED_INTERFACES=YES*** parameter

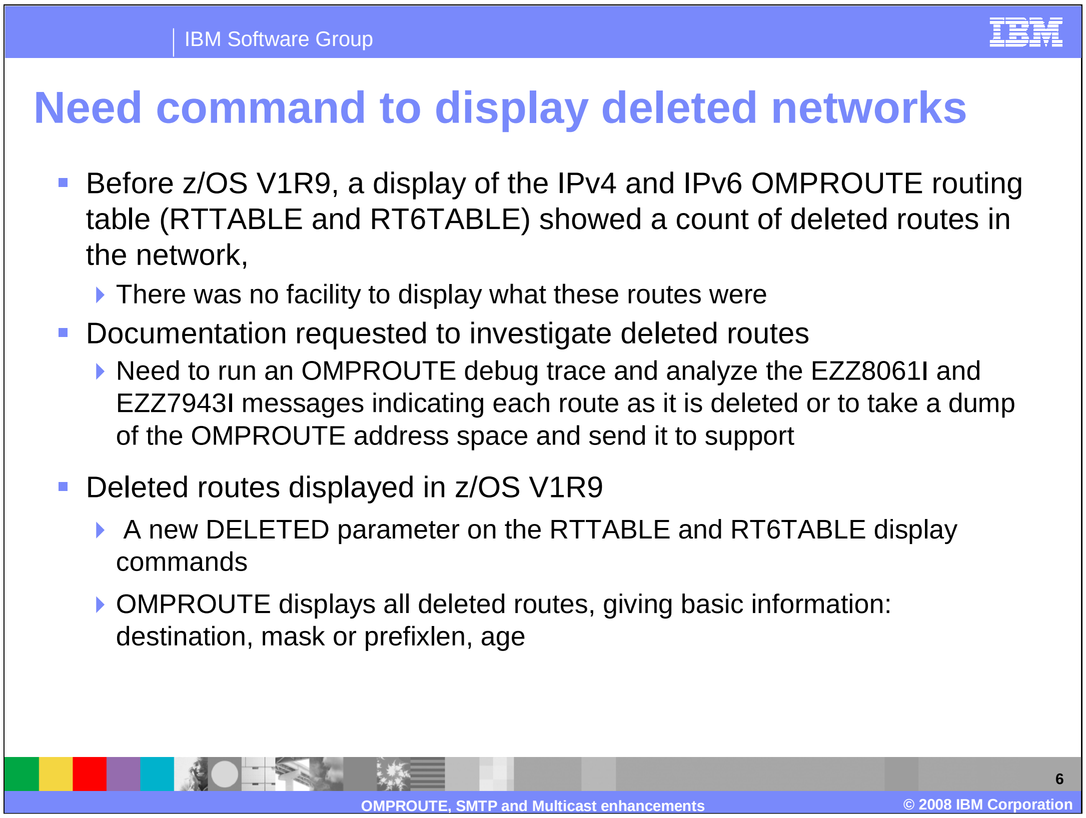
The EZZ7975I *jobname* ignoring undefined interface *interface* message is seen when the GLOBAL_OPTIONS Ignored_Undefined_Interfaces=YES parameter is configured in the OMPROUTE configuration file, and the interface *interface* learned from the TCPIP stack is not defined (or not properly defined) as an IPv4 or IPv6 Interface, RIP_Interface, or OSPF_Interface (as appropriate) in the OMPROUTE configuration file. As a result, OMPROUTE will not update the BSDROUTINGPARMs for the interface in the stack; rather the default MTU of 576 and class mask will be used for IPv4 interfaces which are ignored. (This is not an issue for IPv6 Interfaces). Also, neither the home address (IPv4 and IPv6) or the subnet (IPv4 only) will be advertised through the routing protocol, OMPROUTE will not add a direct route to the interface subnet (IPv4 only), and static routes which use this interface will not be accepted from TCP/IP and therefore will not be advertised (both IPv4 and IPv6).

Service has seen instances where routing problems are caused because users fail to define their interfaces to OMPROUTE. Many routing problems seen in the support stream are caused because of either improperly defined interfaces, or ignored interfaces. These include incorrectly or inadvertently advertised routes or the appearance that expected routes are not being advertised. The problems can be difficult to diagnose without gathering documentation.

Because so many problems have been reported because of improperly defined or undefined interfaces, the EZZ7975I warning message will be output to the console starting in z/OS V1R9. This message will be seen for any interfaces which are defined to TCPIP that do not have a properly coded matching OMPROUTE interface definition if GLOBAL_OPTIONS Ignore_Undefined_Interfaces=YES is coded. This will draw greater attention to these interfaces so that users who have ignored interfaces that they did not intend to ignore will recognize their error more quickly.

If you are intentionally using GLOBAL_OPTIONS Ignore_Undefined_Interfaces=YES to not advertise routing information about particular interfaces, and the interface cited in the message text is one of those interfaces you do not want to advertise, then this message does not indicate a problem. However, if you do want the IP address of the interface cited in the EZZ7975I message to be advertised by the routing protocol to other routers, you will need to either add or correct the definition for this interface in the OMPROUTE configuration file.

The recommendation stands that either all interfaces should be defined to OMPROUTE OR if GLOBAL_OPTIONS Ignore_Undefined_Interface=YES, care must be taken to ensure that any interfaces that are ignored should be ignored and vice versa.



IBM Software Group

Need command to display deleted networks

- Before z/OS V1R9, a display of the IPv4 and IPv6 OMPROUTE routing table (RTTABLE and RT6TABLE) showed a count of deleted routes in the network,
  - There was no facility to display what these routes were
- Documentation requested to investigate deleted routes
  - Need to run an OMPROUTE debug trace and analyze the EZZ8061I and EZZ7943I messages indicating each route as it is deleted or to take a dump of the OMPROUTE address space and send it to support
- Deleted routes displayed in z/OS V1R9
  - A new DELETED parameter on the RTTABLE and RT6TABLE display commands
  - OMPROUTE displays all deleted routes, giving basic information: destination, mask or prefixlen, age

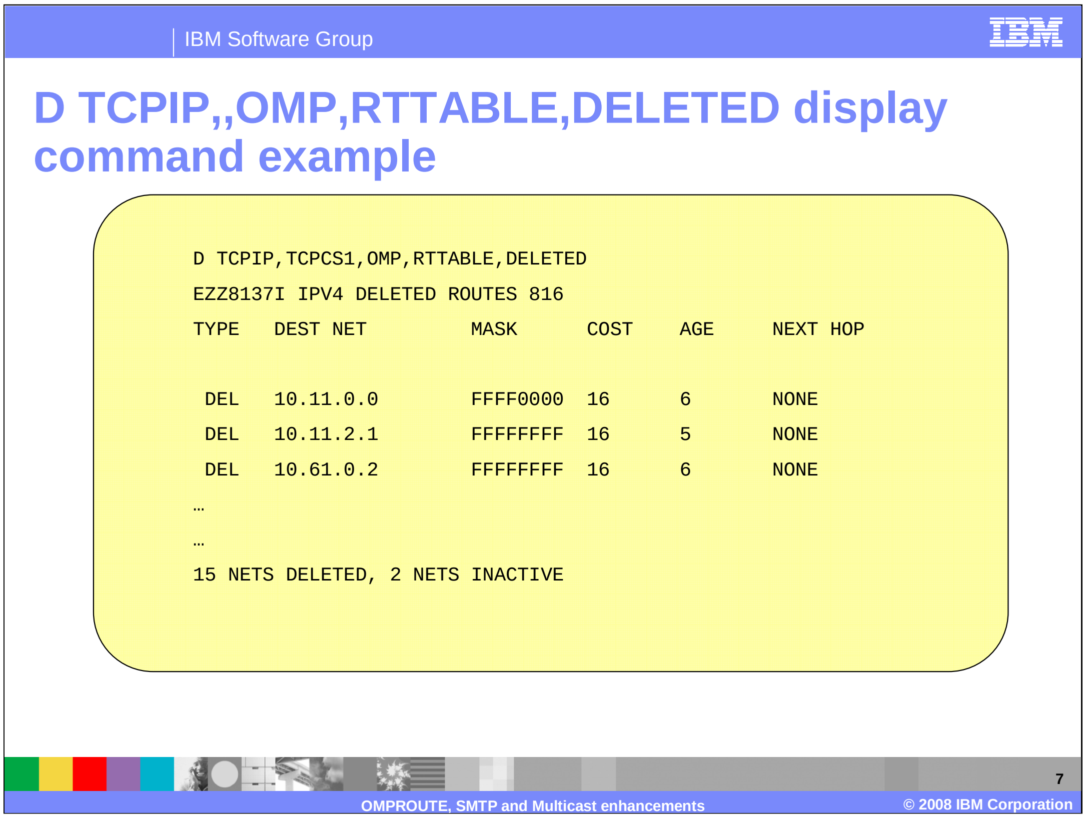
6

OMPROUTE, SMTP and Multicast enhancements

Prior to z/OS V1R9, the OMPROUTE RTTABLE and RT6TABLE displays showed a count of deleted routes, but there was no way to see what these routes actually were.

Because there was no way to display which routes were deleted, you needed to either run an OMPROUTE debug trace and individually find the entries indicating that a route had been deleted, or provide level 2 with a dump of the OMPROUTE address space to extract these routes.

To ease the ability to see the deleted routes, the capability to display deleted routes has been added to the existing RTTABLE and RT6TABLE displays. In z/OS V1R9, a new DELETED parameter has been added to the RTTABLE and RT6TABLE display and modify commands. This parameter can also be used with the new policy based routing OMPROUTE RTTABLE displays for IPv4.

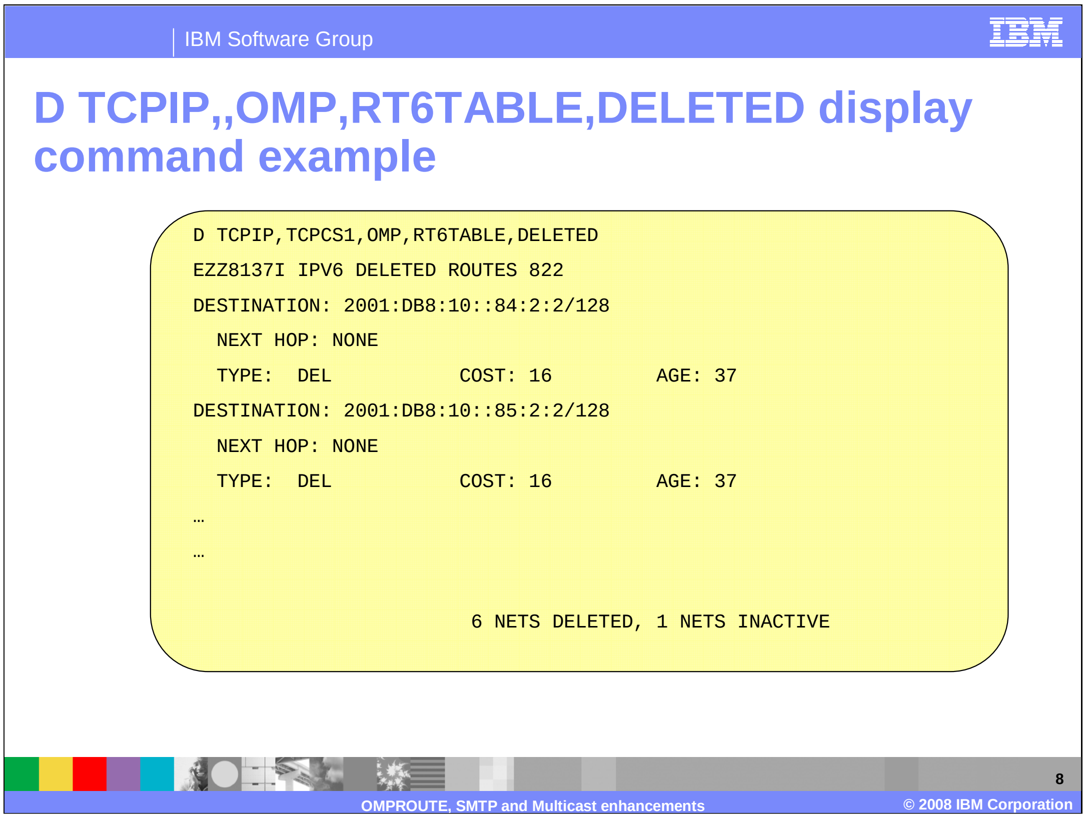


D TCPIP,,OMP,RTTABLE,DELETED display command example

```
D TCPIP,TCPCS1,OMP,RTTABLE,DELETED
EZZ8137I IPV4 DELETED ROUTES 816
TYPE   DEST NET         MASK      COST    AGE     NEXT HOP

 DEL   10.11.0.0        FFFF0000  16      6       NONE
 DEL   10.11.2.1        FFFFFFFF  16      5       NONE
 DEL   10.61.0.2        FFFFFFFF  16      6       NONE
…
…
15 NETS DELETED, 2 NETS INACTIVE
```

This is an example of the output of the D TCPIP,OMP,RTTABLE,DELETED command. This command can be issued to see all deleted routes in OMPROUTE’s main IPv4 routing table. The same information can also be seen in the F OMP,RTTABLE,DELETED display.

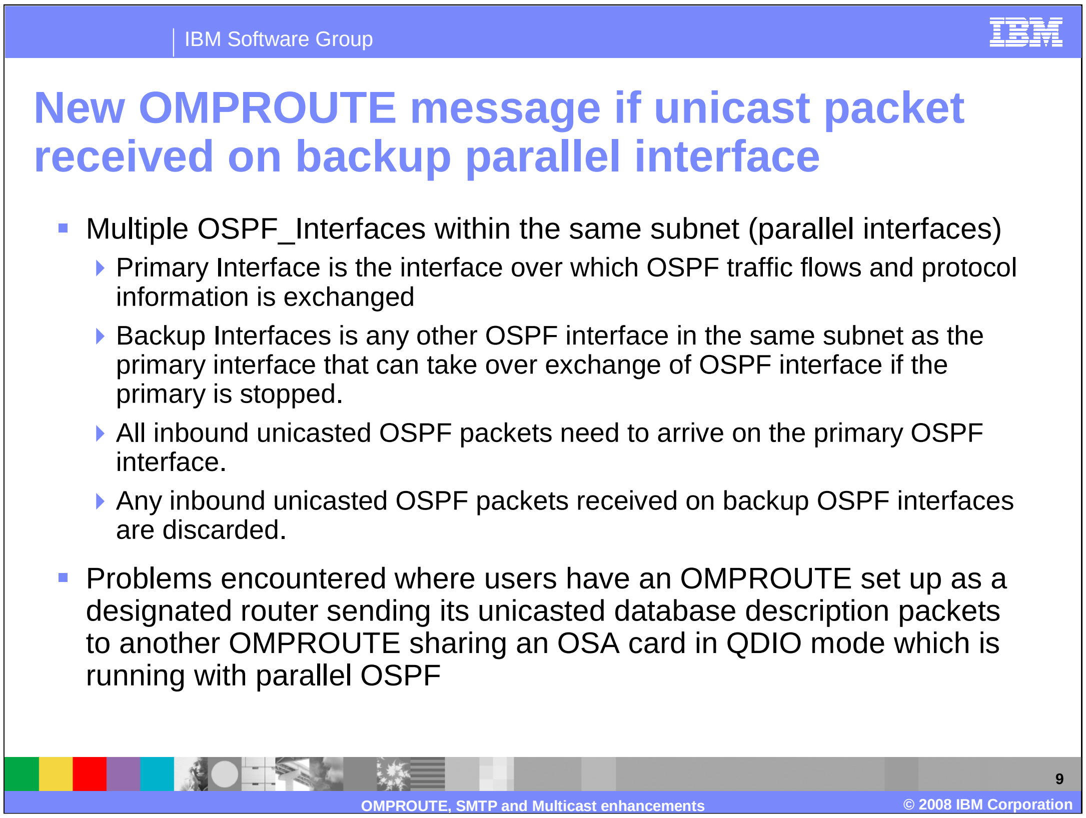


D TCPIP,,OMP,RT6TABLE,DELETED display command example

```
D TCPIP,TCPCS1,OMP,RT6TABLE,DELETED
EZZ8137I IPV6 DELETED ROUTES 822
DESTINATION: 2001:DB8:10::84:2:2/128
  NEXT HOP: NONE
  TYPE:  DEL           COST: 16         AGE: 37
DESTINATION: 2001:DB8:10::85:2:2/128
  NEXT HOP: NONE
  TYPE:  DEL           COST: 16         AGE: 37
…
…

                        6 NETS DELETED, 1 NETS INACTIVE
```

This is an example of the output of the D TCPIP,OMP,RT6TABLE,DELETED command. This command can be issued to see all deleted routes in OMPROUTE’s main IPv6 routing table. The same output can be seen in the F OMPROUTE,RT6TABLE,DELETED display.

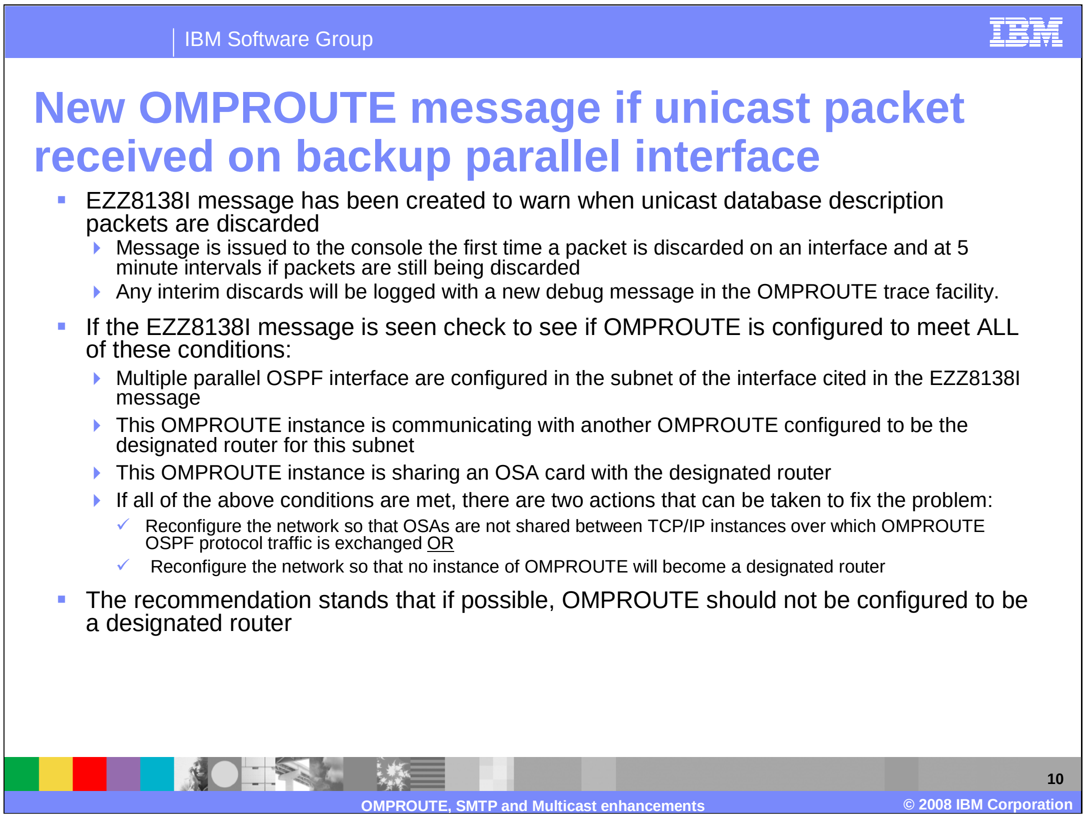


New OMPROUTE message if unicast packet received on backup parallel interface

- Multiple OSPF_Interfaces within the same subnet (parallel interfaces)
  - Primary Interface is the interface over which OSPF traffic flows and protocol information is exchanged
  - Backup Interfaces is any other OSPF interface in the same subnet as the primary interface that can take over exchange of OSPF interface if the primary is stopped.
  - All inbound unicasted OSPF packets need to arrive on the primary OSPF interface.
  - Any inbound unicasted OSPF packets received on backup OSPF interfaces are discarded.
- Problems encountered where users have an OMPROUTE set up as a designated router sending its unicasted database description packets to another OMPROUTE sharing an OSA card in QDIO mode which is running with parallel OSPF

When OMPROUTE is started, if two or more OSPF_Interfaces (or IPv6_OSPF_Interfaces) are in the same subnet (or on the same link, for IPv6), then one of those interfaces will be chosen as the primary OSPF interface. A primary interface can be assigned at the start of OMPROUTE using the Parallel_OSPF=Primary parameter on the OSPF_Interface and IPv6_OSPF_Interface statements, otherwise one will be chosen by OMPROUTE as the primary. The primary OSPF interface is the interface over which the actual OSPF protocol traffic will flow. All other interfaces in that subnet (or on that link) will be given backup status, which means if the primary interface is taken down, one of the backups will take over the responsibility of sending and receiving OSPF protocol traffic. There is a phase of the process of forming an adjacency where OSPF unicast packets called database description packets are sent and received between OMPROUTE and the designated router, and the OSPF protocol specification requires that OMPROUTE receive these packets on the primary OSPF interface. Any packets received on a backup OSPF interface will be discarded.

Service has seen several reported problems where an OMPROUTE with parallel OSPF interfaces is exchanging database description packets with another OMPROUTE which is the designated router, and these two OMPROUTEs share an OSA card in QDIO mode between them. Shared QDIO is wonderful because there is no need to send a packet out into the network if you can just send it to its destination over the OSA card. When the packet is received over the card it is given a “destination address” and then passed up to TCPIP and OMPROUTE. The problem is that this “destination IP address” may or may not be the actual IP address to which the sender sent the packet originally. Most applications do not care what interface a packet is received on, but OMPROUTE does, and if the wrong destination address is assigned by the receiver of a packet learned over the OSA card, and that wrong destination address is for one of the backup parallel ospf interfaces, the packet will be discarded. This can cause neighbor state regressions and loops, or if an adjacency had previously formed, adjacency failures can occur, resulting in the loss of routes.



IBM Software Group

New OMPROUTE message if unicast packet received on backup parallel interface

- EZZ8138I message has been created to warn when unicast database description packets are discarded
  - Message is issued to the console the first time a packet is discarded on an interface and at 5 minute intervals if packets are still being discarded
  - Any interim discards will be logged with a new debug message in the OMPROUTE trace facility.
- If the EZZ8138I message is seen check to see if OMPROUTE is configured to meet ALL of these conditions:
  - Multiple parallel OSPF interface are configured in the subnet of the interface cited in the EZZ8138I message
  - This OMPROUTE instance is communicating with another OMPROUTE configured to be the designated router for this subnet
  - This OMPROUTE instance is sharing an OSA card with the designated router
  - If all of the above conditions are met, there are two actions that can be taken to fix the problem:
    - ✓ Reconfigure the network so that OSAs are not shared between TCP/IP instances over which OMPROUTE OSPF protocol traffic is exchanged OR
    - ✓ Reconfigure the network so that no instance of OMPROUTE will become a designated router
- The recommendation stands that if possible, OMPROUTE should not be configured to be a designated router

10

OMPROUTE, SMTP and Multicast enhancements

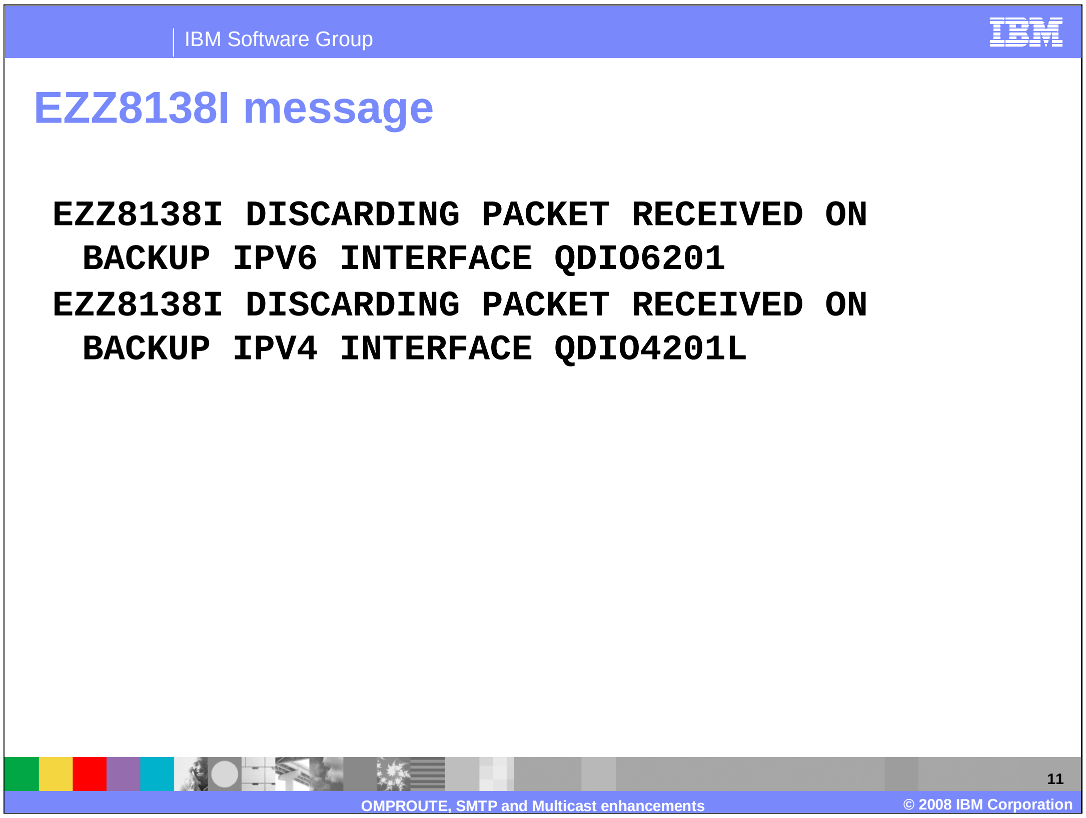
© 2008 IBM Corporation

A new message, EZZ8138I, has been created to output to the console the first time a packet is received on a backup parallel ospf interface and discarded. After the first time this message is seen, it will be suppressed and only be issued to the console every 5 minutes to prevent flooding. Any interim discards are logged in an OMPROUTE trace if one is running.

This message will aid in diagnosis of the problem cited above You can only be experiencing this particular problem if you meet all the criteria; 1) multiple parallel OSPF interfaces in the same subnet (or link for IPv6), 2) the designated router for this subnet is another OMPROUTE instance, and 3) this OMPROUTE is sharing an OSA card in QDIO mode with the designated router OMPROUTE. If this is your problem, there are two ways to fix it: 1) reconfigure the network so that the OSAs are not shared between the designated router OMPROUTE's TCPIP stack and any other OMPROUTE's TCPIP stacks, or 2) reconfigure the network so that no instance of OMPROUTE will be the designated router (if possible).

If you do not meet the criteria for being in the shared QDIO problem, but you are still seeing EZZ8138I messages, then it would appear that some other problem is occurring or some other router is sending unsolicited OSPF protocol traffic to one or more of your backup parallel interfaces. In that case, you may want to check to see if packets are being leaked across VLANs, if those are in use. If not, the documentation that needs to be gathered is a –t2 –d1 (or –6t2 -6d3 for IPv6 OSPF) trace and a dump of both OMPROUTE and TCPIP's address spaces.

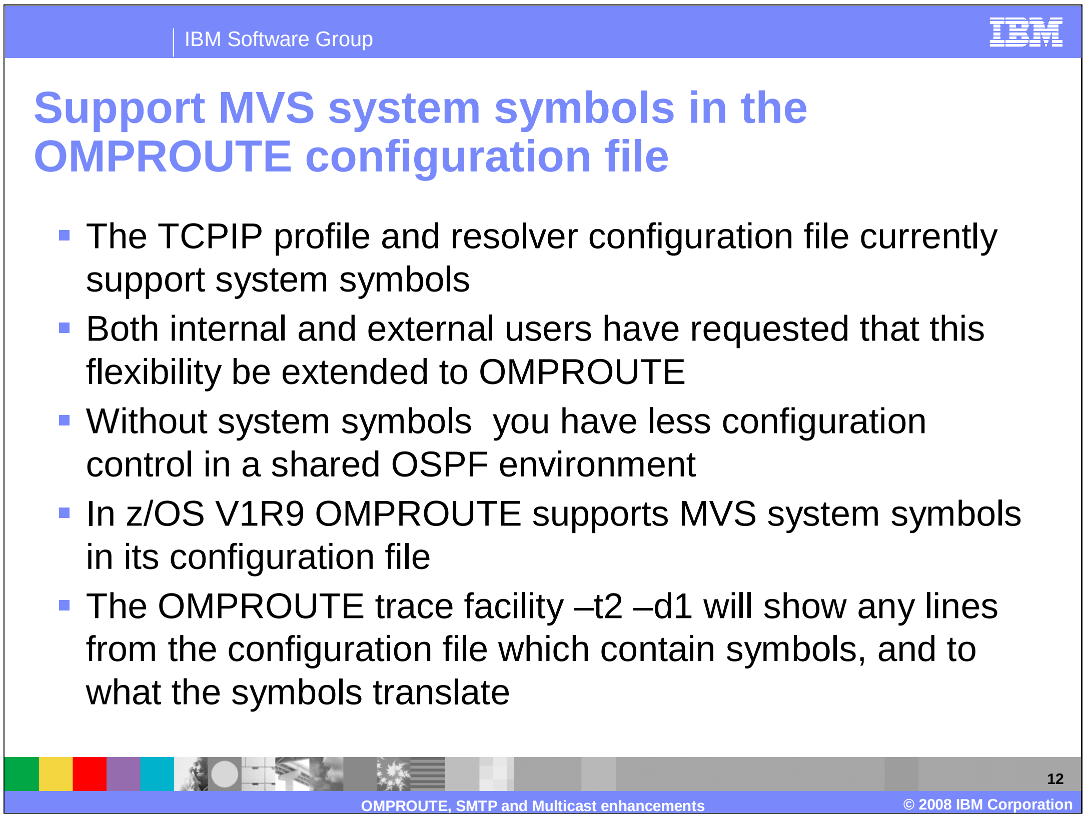
If possible, do not let OMPROUTE become the designated router for the subnet or link. You can configure the Router_Priority=0 parameter on the OSPF_Interface or IPv6_OSPF_Interface statement to keep an instance of OMPROUTE from becoming a designated router. Be careful though – if you ONLY have OMPROUTE instances in your subnet, one of them then HAS to be a designated router. In that case, chose that designated router so that you can avoid sharing an OSA between it and any other OMPROUTE in that subnet.



EZZ8138I message

```
EZZ8138I DISCARDING PACKET RECEIVED ON
  BACKUP IPV6 INTERFACE QDIO6201
EZZ8138I DISCARDING PACKET RECEIVED ON
  BACKUP IPV4 INTERFACE QDIO4201L
```

Here are two examples of what the new EZZ8138I message looks like for an IPv4 and IPv6 OSPF interface.



Support MVS system symbols in the OMPROUTE configuration file

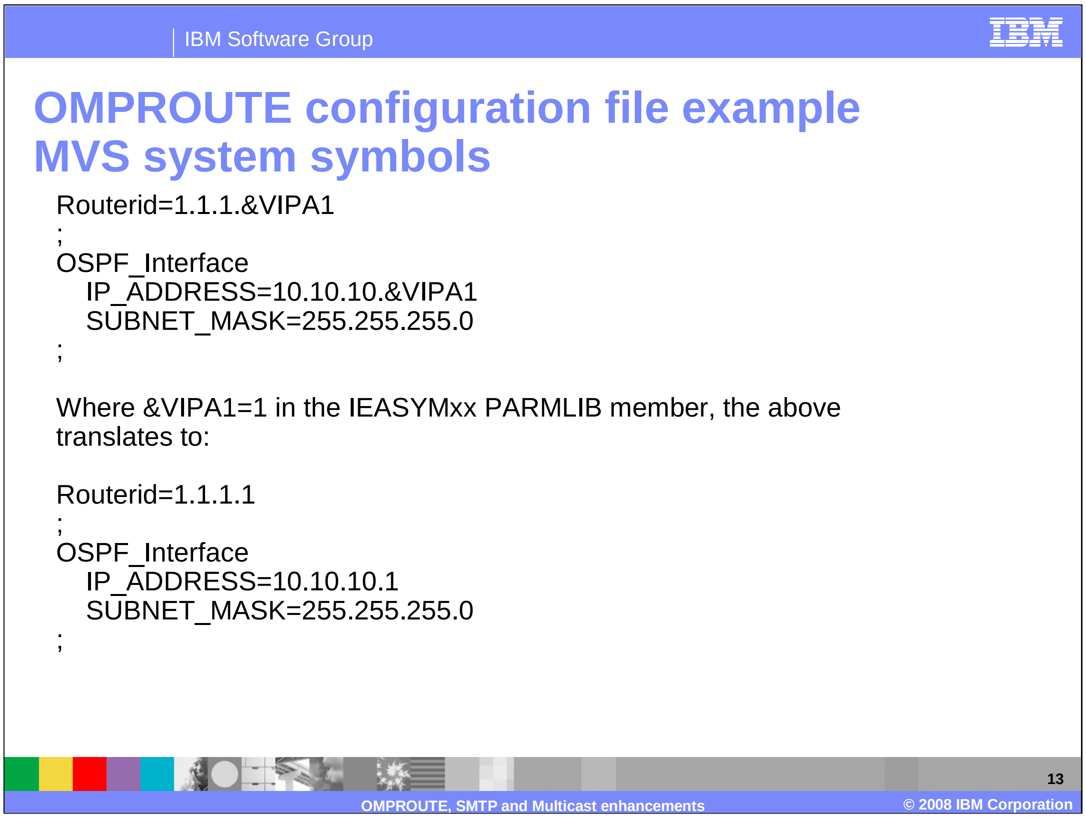
- The TCPIP profile and resolver configuration file currently support system symbols
- Both internal and external users have requested that this flexibility be extended to OMPROUTE
- Without system symbols you have less configuration control in a shared OSPF environment
- In z/OS V1R9 OMPROUTE supports MVS system symbols in its configuration file
- The OMPROUTE trace facility –t2 –d1 will show any lines from the configuration file which contain symbols, and to what the symbols translate

Prior to V1R9, the TCPIP Profile and Resolver configuration files supported MVS system symbols, however OMPROUTE did not. Both internal and external users have been interested in adding this functionality.

The ability to use the MVS system symbols in the OMPROUTE configuration file is nice in and of itself because now OMPROUTE configuration files can be shared between OMPROUTE instances. It was possible to share configuration files between OMPROUTE instances before V1R9 by using wildcarding; however in an OSPF environment there was no way to wildcard the Routerid, so if you did share configuration files, there was no way to specify a unique routerid for each OMPROUTE instance.

OMPROUTE now supports MVS system symbols in its configuration files.

If you need to see how a symbol was translated, turn on –t2 –d1 OMPROUTE trace and look for the text “Translated to”. For each line that contained an MVS system symbol there will be a line in the trace file which shows to what the symbol was translated.



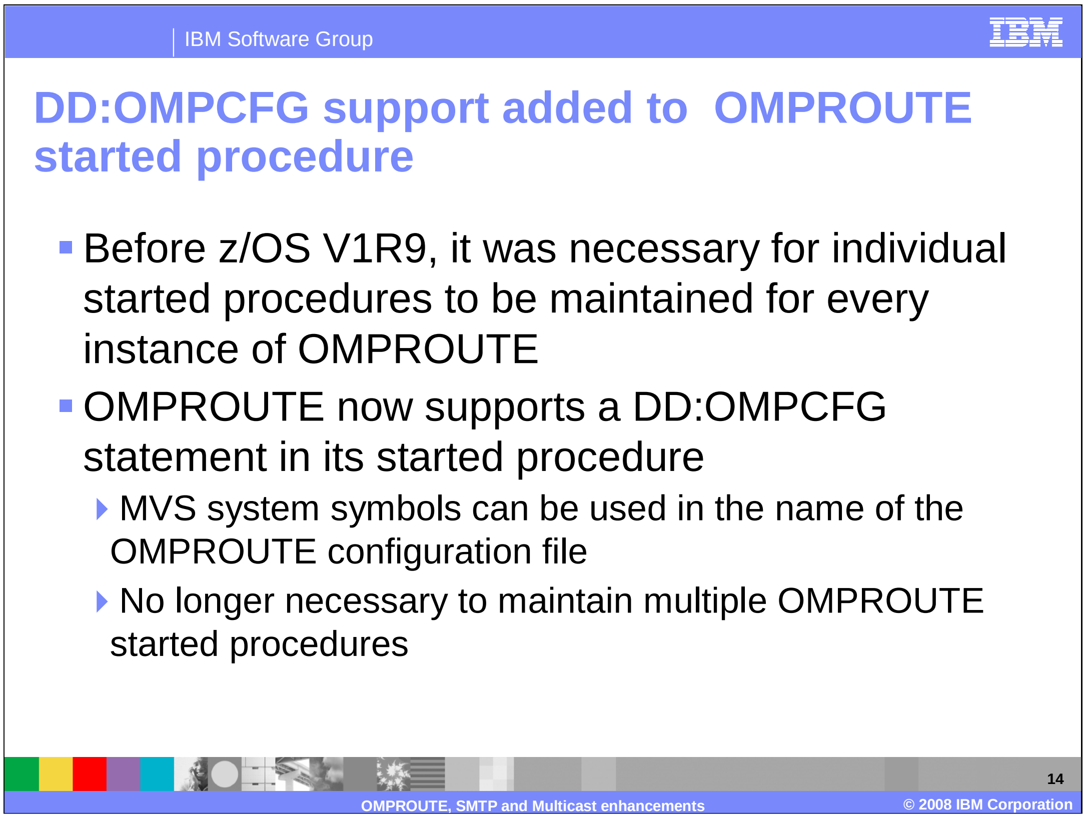
OMPROUTE configuration file example MVS system symbols

```
Routerid=1.1.1.&VIPA1
;
OSPF_Interface
   IP_ADDRESS=10.10.10.&VIPA1
   SUBNET_MASK=255.255.255.0
;
```

Where &VIPA1=1 in the IEASYMxx PARMLIB member, the above translates to:

```
Routerid=1.1.1.1
;
OSPF_Interface
   IP_ADDRESS=10.10.10.1
   SUBNET_MASK=255.255.255.0
;
```

This is an example of how symbol translation can be used in the OMPROUTE configuration file.

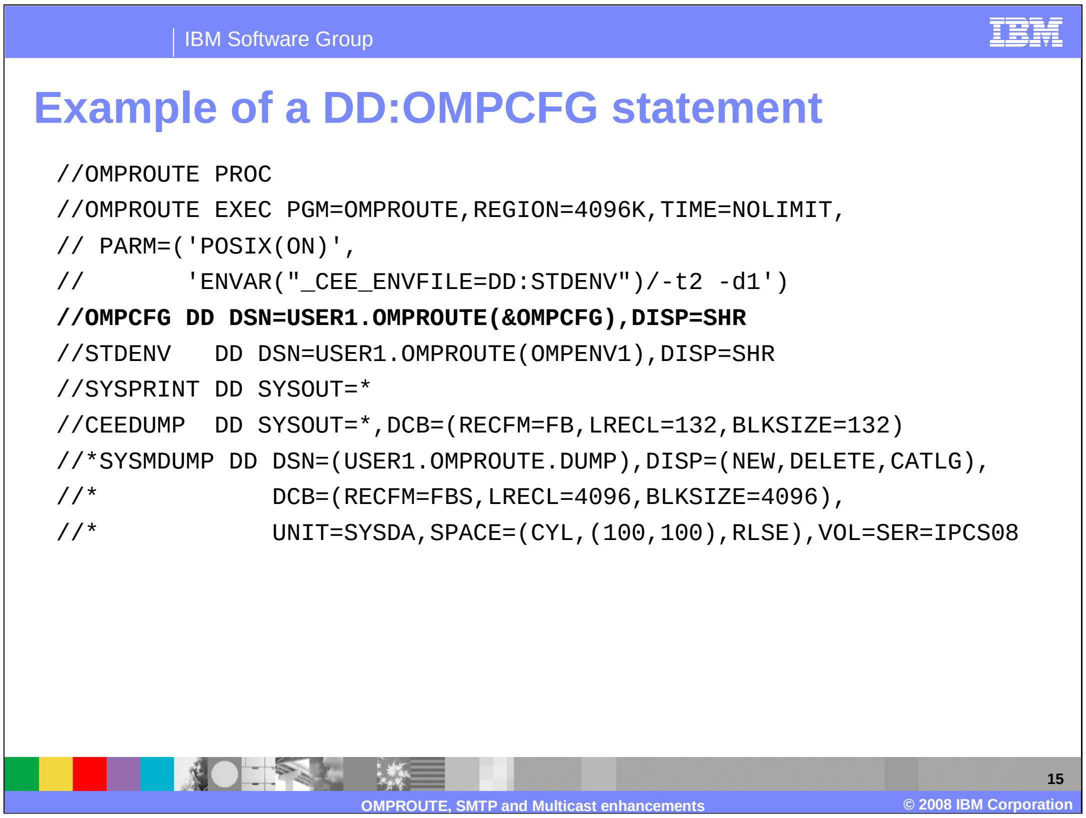


DD:OMPCFG support added to OMPROUTE started procedure

- Before z/OS V1R9, it was necessary for individual started procedures to be maintained for every instance of OMPROUTE
- OMPROUTE now supports a DD:OMPCFG statement in its started procedure
  - MVS system symbols can be used in the name of the OMPROUTE configuration file
  - No longer necessary to maintain multiple OMPROUTE started procedures

Prior to z/OS V1R9 it was necessary for individual started procedures to be maintained for each OMPROUTE instance.

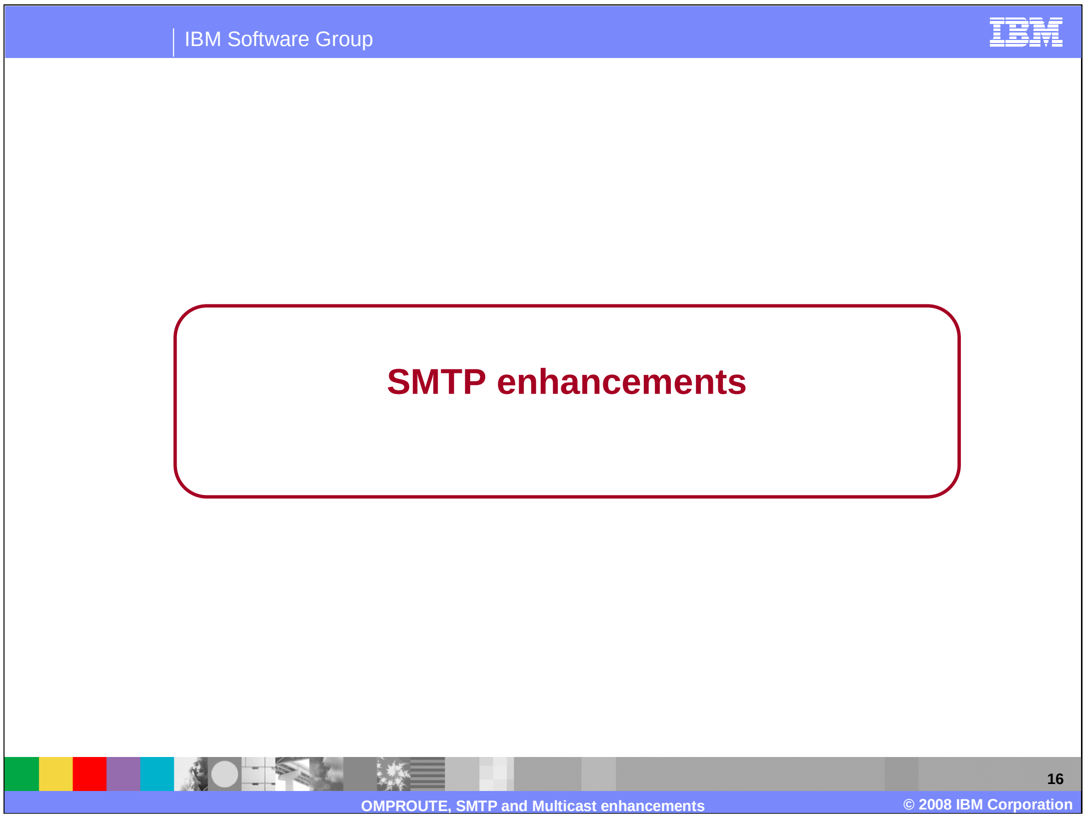
Both internal and external users have requested a way to specify an OMPROUTE configuration file name which includes an MVS system symbol in the started procedure for OMPROUTE, so that one started procedure could be shared by multiple OMPROUTE instances.
OMPROUTE now supports a DD:OMPCFG statement in its started procedure. This allows for MVS system symbols to be used in the name of the OMPROUTE configuration file, eliminating the necessity to maintain multiple OMPROUTE started procedures



Example of a DD:OMPCFG statement

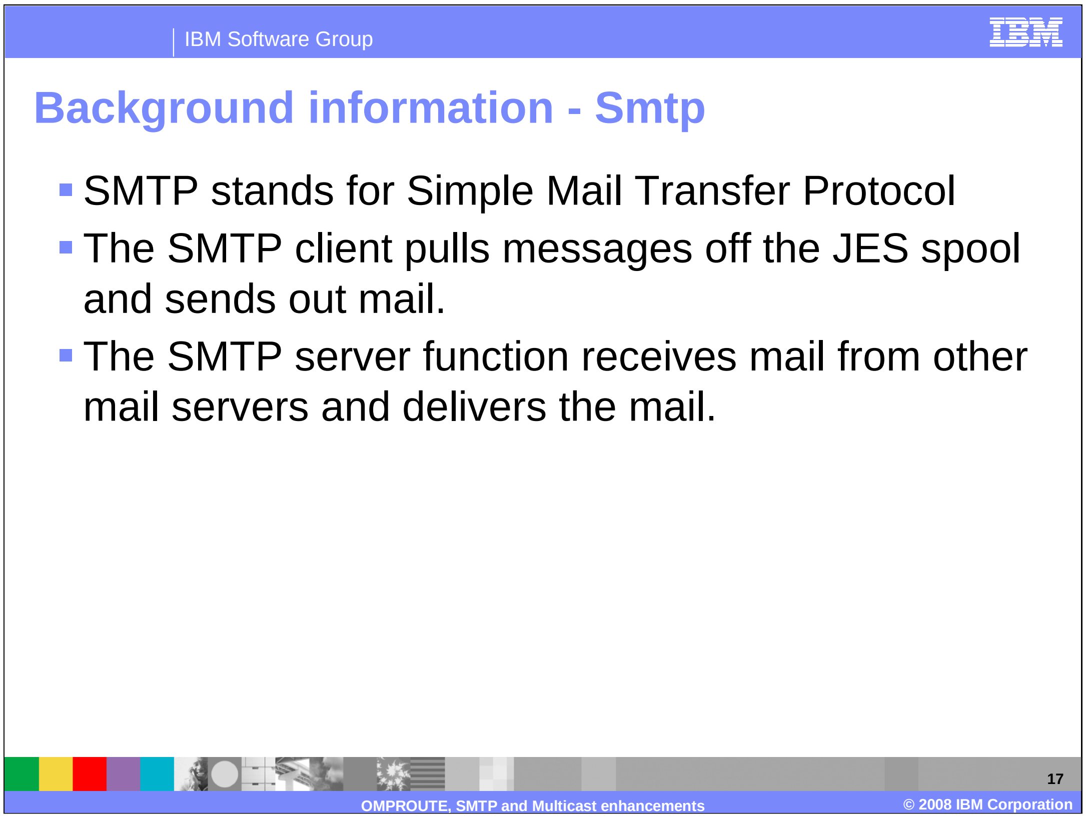
```
//OMPROUTE PROC
//OMPROUTE EXEC PGM=OMPROUTE,REGION=4096K,TIME=NOLIMIT,
// PARM=('POSIX(ON)',
//       'ENVAR("_CEE_ENVFILE=DD:STDENV")/-t2 -d1')
//OMPCFG DD DSN=USER1.OMPROUTE(&OMPCFG),DISP=SHR
//STDENV   DD DSN=USER1.OMPROUTE(OMPENV1),DISP=SHR
//SYSPRINT DD SYSOUT=*
//CEEDUMP  DD SYSOUT=*,DCB=(RECFM=FB,LRECL=132,BLKSIZE=132)
//*SYSMDUMP DD DSN=(USER1.OMPROUTE.DUMP),DISP=(NEW,DELETE,CATLG),
//*            DCB=(RECFM=FBS,LRECL=4096,BLKSIZE=4096),
//*            UNIT=SYSDA,SPACE=(CYL,(100,100),RLSE),VOL=SER=IPCS08
```

This is an example of how the new OMPCFG DD can be used. USER1.OMPROUTE(&OMPCFG) is the name of the OMPROUTE configuration file, where &OMPCFG is an MVS system symbol defined in the IEASYSYMxx parmlib member.



SMTP enhancements

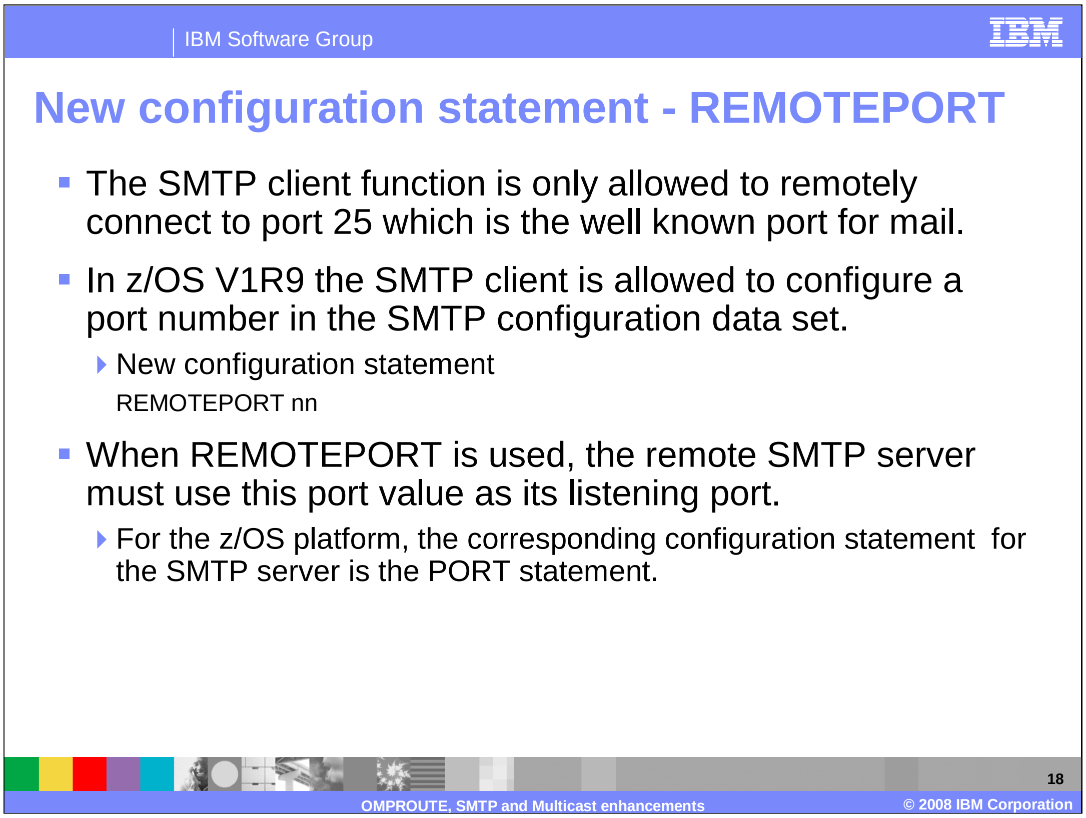
This section covers the enhancements made to SMTP in z/OS V1R9 Communications Server.



Background information - Smtp

- SMTP stands for Simple Mail Transfer Protocol
- The SMTP client pulls messages off the JES spool and sends out mail.
- The SMTP server function receives mail from other mail servers and delivers the mail.

Here is some background information on SMTP. SMTP stands for Simple Mail Transfer Protocol. It supports RFCs 821 and 822. SMTP is used to pull messages off the JES spool. Messages are put on the spool using IEBGENER, TSO TRANSMIT, or SMTPNOTE. The SMTP client function is used to send out the mail to either local users or to remote mail servers. The SMTP server function puts a listen up on the default port of 25 and accepts mail from other mail servers or client socket programs. It delivers the mail to local users, or relays the mail to remote mail servers.



IBM Software Group

New configuration statement - REMOTEPORT

- The SMTP client function is only allowed to remotely connect to port 25 which is the well known port for mail.
- In z/OS V1R9 the SMTP client is allowed to configure a port number in the SMTP configuration data set.
  - New configuration statement

    REMOTEPORT nn
- When REMOTEPORT is used, the remote SMTP server must use this port value as its listening port.
  - For the z/OS platform, the corresponding configuration statement for the SMTP server is the PORT statement.

18

OMPROUTE, SMTP and Multicast enhancements

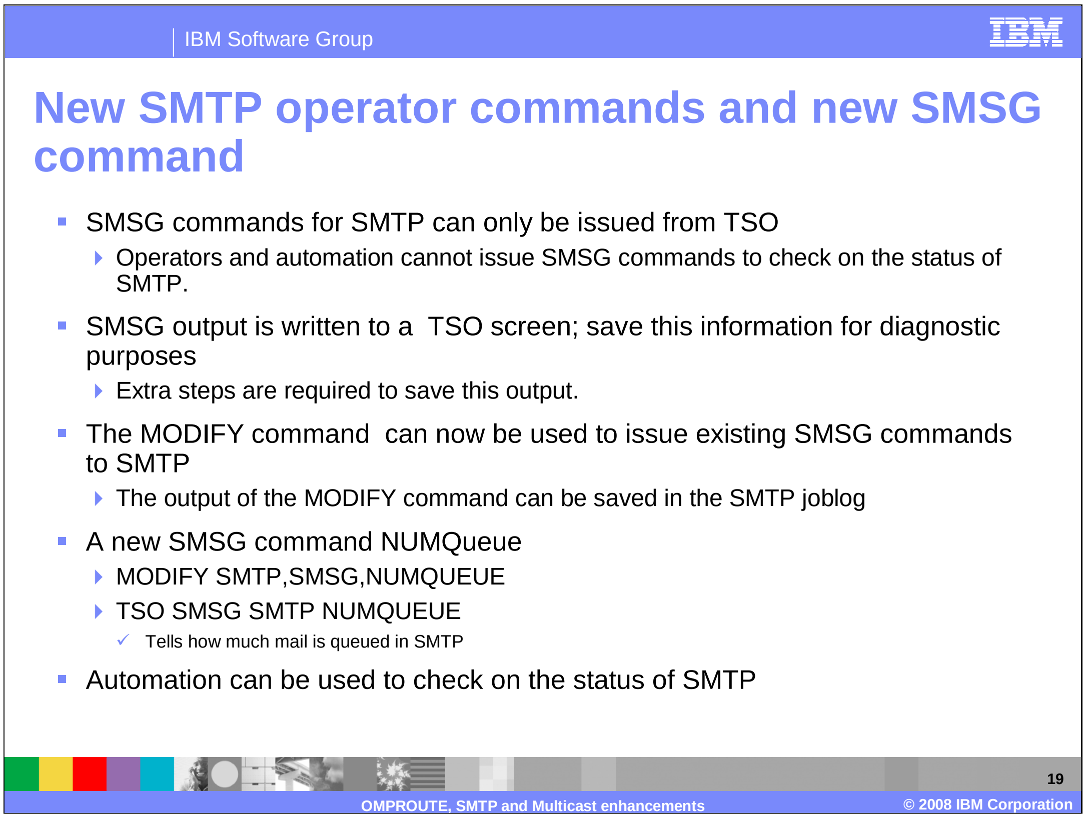
© 2008 IBM Corporation

In earlier releases, the SMTP client function is only allowed to remotely connect to the well known port for mail which is port 25. This is too restrictive. Users would like SMTP to support a configuration option so that the SMTP client function can remotely connect to a configured port value which the system administrator can chose. This configuration can be useful in a testing environment.
So support was added for a new configuration statement so that the SMTP client function can remotely connect to a configured port value which the system administrator can chose. The SMTP started task now supports a new configuration statement REMOTEPORT. This value will be identified at initialization time and used by the SMTP client so that this port number, and not 25, is used during connect processing. As a result, the SMTP client will use this port number as the remote port to send all outbound mail. This value may not be reset dynamically. To change it, you must stop and restart the SMTP started task. Of course if the statement is not coded the default is port 25.

For parameters the value of the nn is a decimal number. This parameter must be within the range of 1 to 65534 and is limited to ten characters.

Note that if the REMOTEPORT statement is coded it must be within the range and there is no default taken if the statement is coded incorrectly. Furthermore, SMTP will not start.

If the statement is coded, then the SMTP client will use this port value to connect to the remote SMTP server. If no SMTP server is listening on that port then mail cannot be delivered. On the z/OS platform the corresponding configuration statement that needs to be modified for the SMTP server is the PORT statement in the SMTP configuration file on the system where SMTP server is started and the mail is to be sent. Also for z/OS platform, the SMTP server port value used should be reserved on the PORT statement in the TCP/IP configuration file as well.



IBM Software Group

New SMTP operator commands and new SMSG command

- SMSG commands for SMTP can only be issued from TSO
  - Operators and automation cannot issue SMSG commands to check on the status of SMTP.
- SMSG output is written to a TSO screen; save this information for diagnostic purposes
  - Extra steps are required to save this output.
- The MODIFY command can now be used to issue existing SMSG commands to SMTP
  - The output of the MODIFY command can be saved in the SMTP joblog
- A new SMSG command NUMQueue
  - MODIFY SMTP,SMSG,NUMQUEUE
  - TSO SMSG SMTP NUMQUEUE
    - ✓ Tells how much mail is queued in SMTP
- Automation can be used to check on the status of SMTP

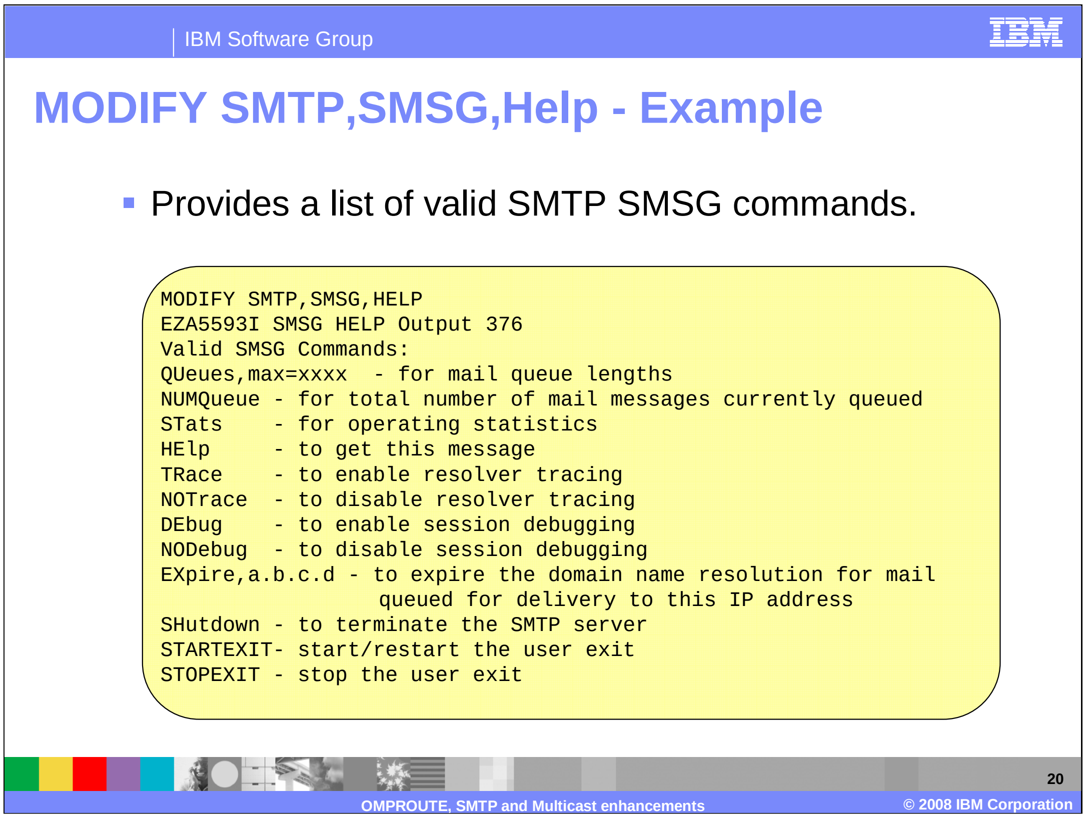
19

OMPROUTE, SMTP and Multicast enhancements

© 2008 IBM Corporation

To monitor SMTP, it would be helpful to have automation or operators check on the status of SMTP. But, since TSO SMSG commands cannot be issued from automation or from operators (if they are not logged on to TSO), this cannot be done. Also, there is not an easy way to save the output of the TSO SMSG commands. So you must take extra steps to save this output.

In z/OS V1R9, support was added for a MODIFY command that can be issued by automation and operators. A new SMSG command NUMQueue was also added to tell how much mail is queued in SMTP. The new command can using the MODIFY command or TSO. The output of the Modify command is saved in the SMTP joblog. Now, automation and operators can check on the status of SMTP.

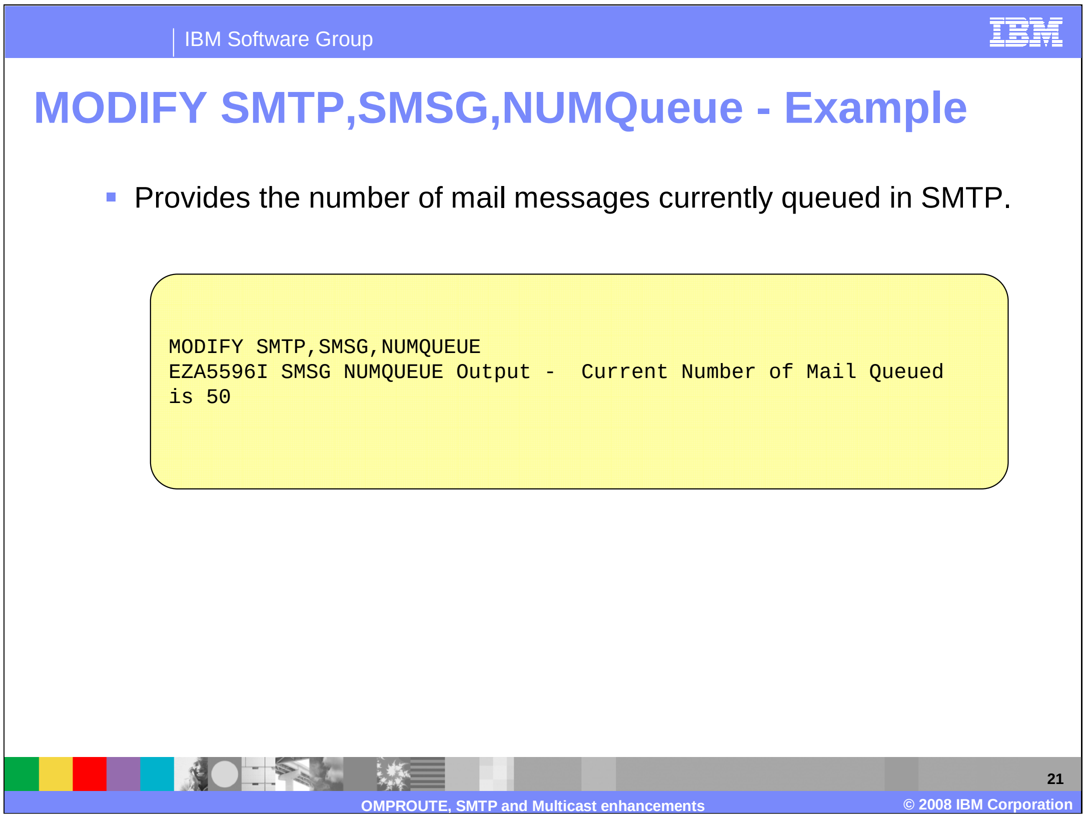


MODIFY SMTP,SMSG,Help - Example

- Provides a list of valid SMTP SMSG commands.

```
MODIFY SMTP,SMSG,HELP
EZA5593I SMSG HELP Output 376
Valid SMSG Commands:
QUeues,max=xxxx  - for mail queue lengths
NUMQueue - for total number of mail messages currently queued
STats    - for operating statistics
HElp     - to get this message
TRace    - to enable resolver tracing
NOTrace  - to disable resolver tracing
DEbug    - to enable session debugging
NODebug  - to disable session debugging
EXpire,a.b.c.d - to expire the domain name resolution for mail
                 queued for delivery to this IP address
SHutdown - to terminate the SMTP server
STARTEXIT- start/restart the user exit
STOPEXIT - stop the user exit
```

The MODIFY SMTP,SMSG,HELP lists the valid MODIFY SMTP SMSG commands.

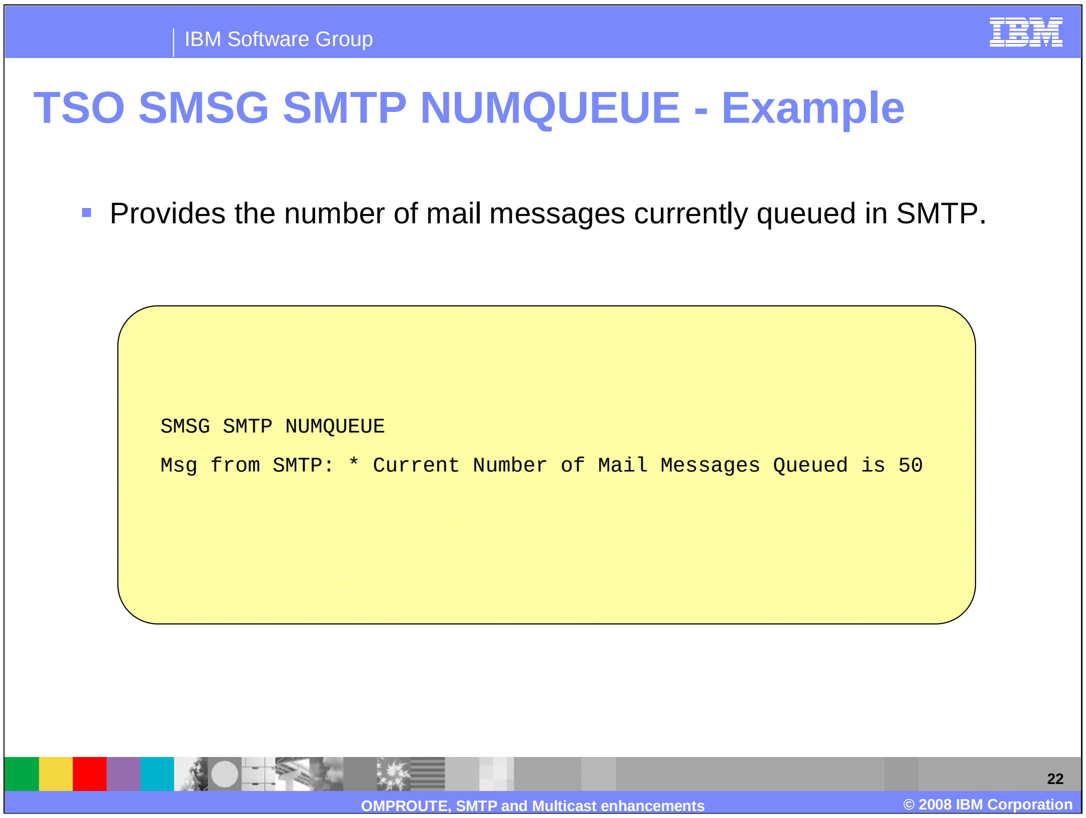


MODIFY SMTP,SMSG,NUMQueue - Example

- Provides the number of mail messages currently queued in SMTP.

```
MODIFY SMTP,SMSG,NUMQUEUE
EZA5596I SMSG NUMQUEUE Output -  Current Number of Mail Queued
is 50
```

The MODIFY SMTP,SMSG,NUMQUEUE command provides the current number of mail messages queued in SMTP.

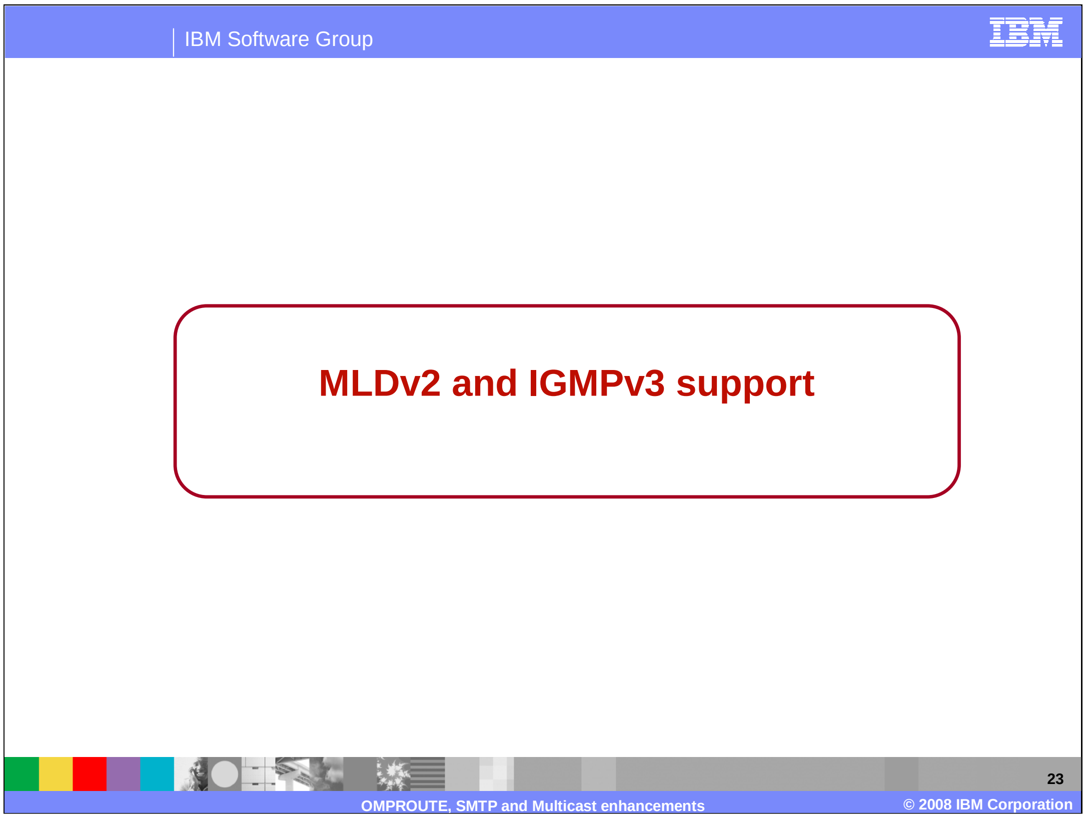


TSO SMSG SMTP NUMQUEUE - Example

- Provides the number of mail messages currently queued in SMTP.

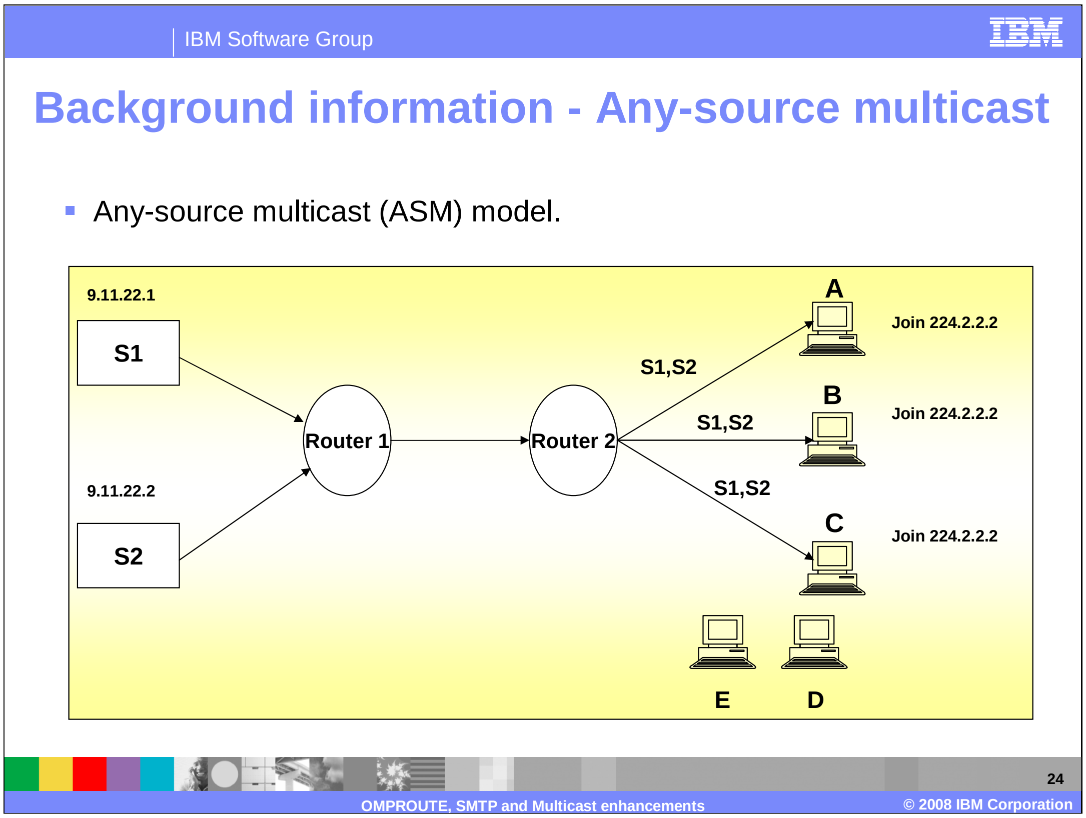
```
SMSG SMTP NUMQUEUE
Msg from SMTP: * Current Number of Mail Messages Queued is 50
```

There is a new SMSG command, NUMQUEUE, that can be issued from TSO to provide the current number of mail messages queued in SMTP.



MLDv2 and IGMPv3 support

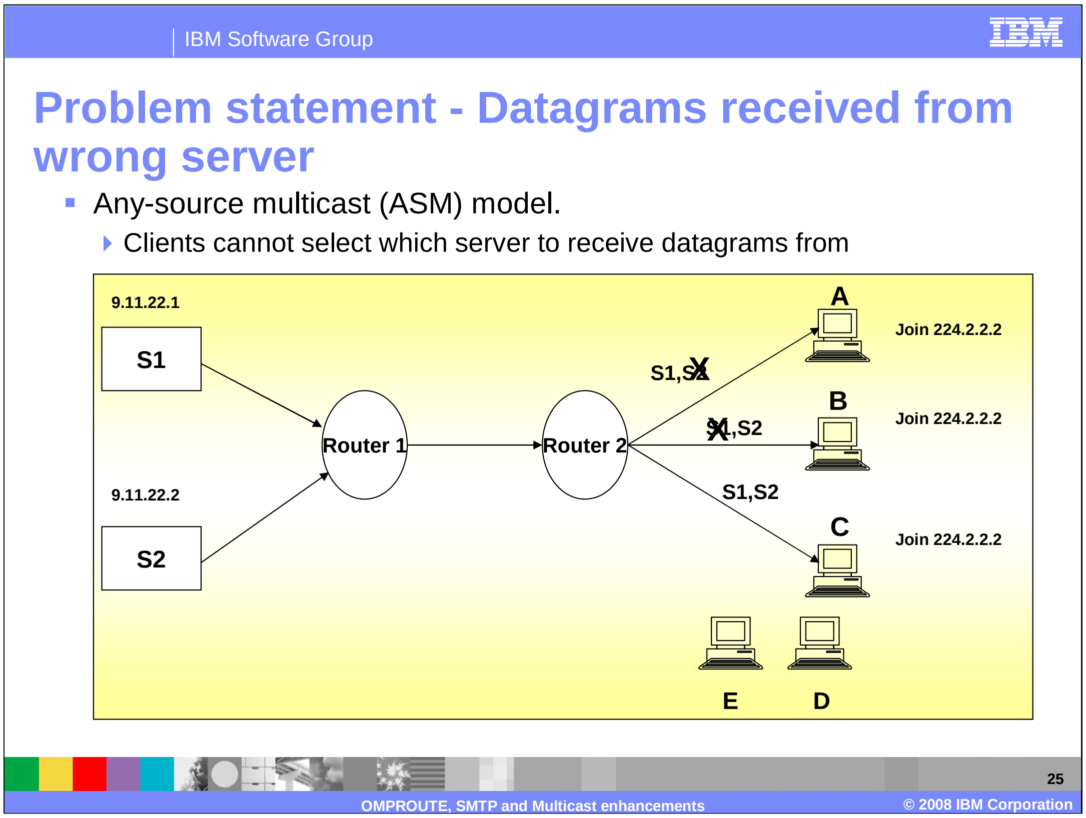
This section covers the implementation of the new versions of the Internet Group Management Protocol for IPv4 , IGMPv3, and Multicast Listening Discovery for IPv6, MLDv2.



Background information - Any-source multicast

- Any-source multicast (ASM) model.

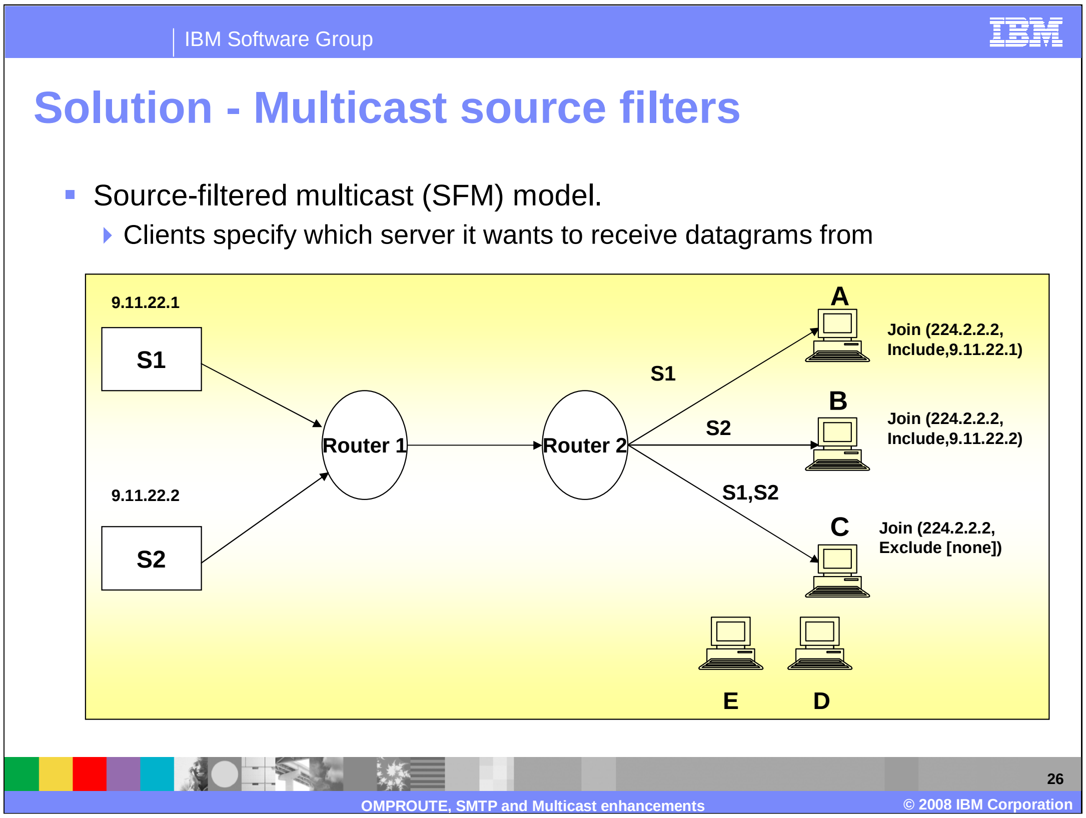
The current multicast model is referred to as any-source multicast. Multicast server programs send out datagrams using a multicast address as the destination address. Any client program on the network can choose to receive the multicast datagrams by joining the multicast group. This means that a client program which has joined a multicast group will receive multicast datagram's from any server, regardless of the source IP address. In the example, clients A, B and C receive all datagram's from both server 1 and server 2. An application could specify which multicast datagrams it wanted to receive by specifying the multicast address as the filter. However all multicast datagrams which met that criteria, regardless of the source address, would be delivered to the application. This is referred to as the Any-Source Multicast (ASM) model.   In the diagram, servers S1 and S2 send datagrams to the multicast group address 224.2.2.2. Clients A, B and C receive datagrams from servers S1 and S2 due to joining the multicast group address 224.2.2.2. Client D and and E do not receive any datagrams from S1 and S2 since they did not join the multicast group address 224.2.2.2.



Problem statement - Datagrams received from wrong server

- Any-source multicast (ASM) model.
  - Clients cannot select which server to receive datagrams from

It’s possible for multiple multicast servers to be sending out different information using the same destination multicast address. With the any-source multicast model, this can cause problems if a client only wants to receive datagram’s from a specific server. For example, if multiple servers are sending out different audio feeds, a client might only want to receive audio from one source. In the example, client A wants to receive datagram’s from server 1 but does not want to receive datagrams from server 2 and client B wants to receive datagram’s from server 2 but not server 1.



IBM Software Group

Solution - Multicast source filters

- Source-filtered multicast (SFM) model.
  - Clients specify which server it wants to receive datagrams from

9.11.22.1
S1
9.11.22.2
S2
Router 1
Router 2
S1
S2
S1,S2
A
Join (224.2.2.2, Include,9.11.22.1)
B
Join (224.2.2.2, Include,9.11.22.2)
C
Join (224.2.2.2, Exclude [none])
E
D

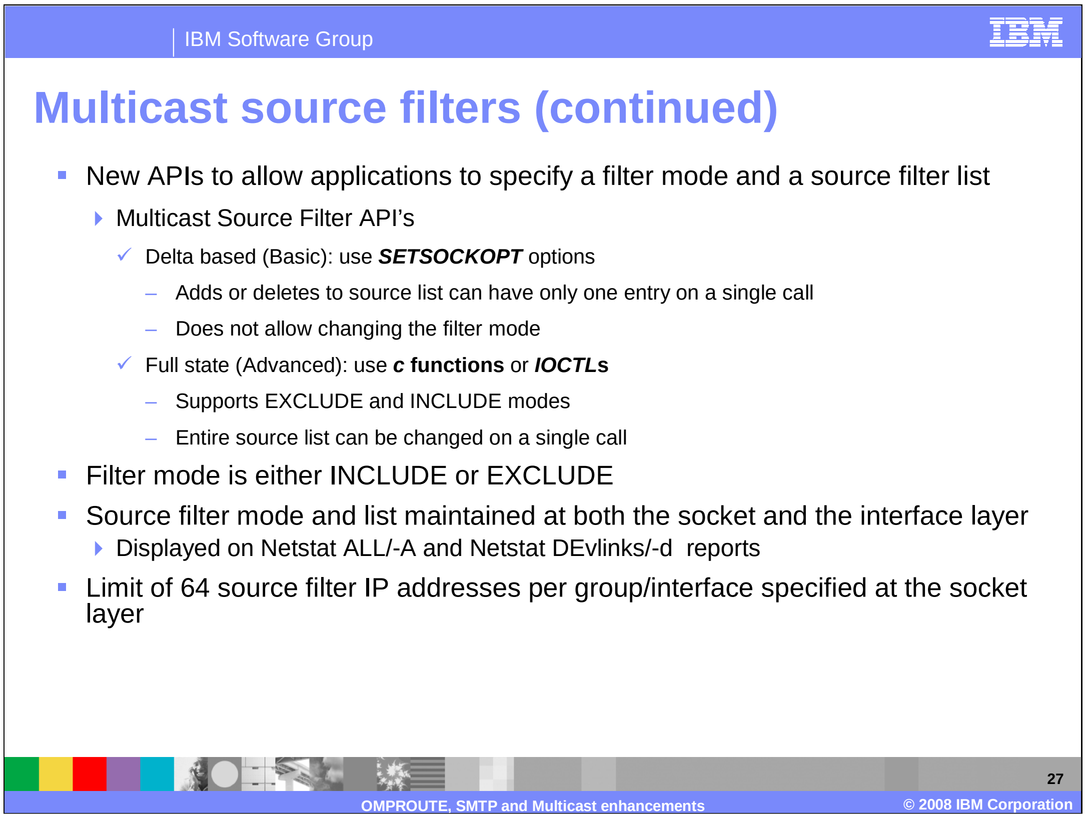
26

OMPROUTE, SMTP and Multicast enhancements

© 2008 IBM Corporation

The solution is to allow a client to specify filters based on the source IP address of the multicast datagram. This model is called source-filtered multicast. In the example, client A can specify that it only wants multicast datagram's from server 1 and client B only wants to receive multicast datagram's from server 2. Client C wants to receive all multicast datagram's, regardless of the source IP address.

The SFM model allows an application to filter the datagrams it receives based on the source IP address. In the diagram servers S1 and S2 send datagrams to the multicast group address 224.2.2.2. Client A receives datagrams from server S1 only due to joining the multicast group address 224.2.2.2 with the source filter mode of INCLUDE for the source IP address 9.11.22.1. Client B receives datagrams from server S2 only due to joining the multicast group address 224.2.2.2 with the source filter mode of INCLUDE for the source IP address 9.11.22.2. Client C receives datagrams from servers S1 and S2 due to joining the multicast group address 224.2.2.2 with the source filter mode of EXCLUDE with an empty source list. Clients D and E do not receive any datagram from S1 and S2 because they did not join the multicast group address 224.2.2.2.



IBM Software Group

Multicast source filters (continued)

- New APIs to allow applications to specify a filter mode and a source filter list
  - Multicast Source Filter API's
    - ✓ Delta based (Basic): use ***SETSOCKOPT*** options
      - Adds or deletes to source list can have only one entry on a single call
      - Does not allow changing the filter mode
    - ✓ Full state (Advanced): use ***c* functions** or ***IOCTLs***
      - Supports EXCLUDE and INCLUDE modes
      - Entire source list can be changed on a single call
- Filter mode is either INCLUDE or EXCLUDE
- Source filter mode and list maintained at both the socket and the interface layer
  - Displayed on Netstat ALL/-A and Netstat DEvlinks/-d reports
- Limit of 64 source filter IP addresses per group/interface specified at the socket layer

27

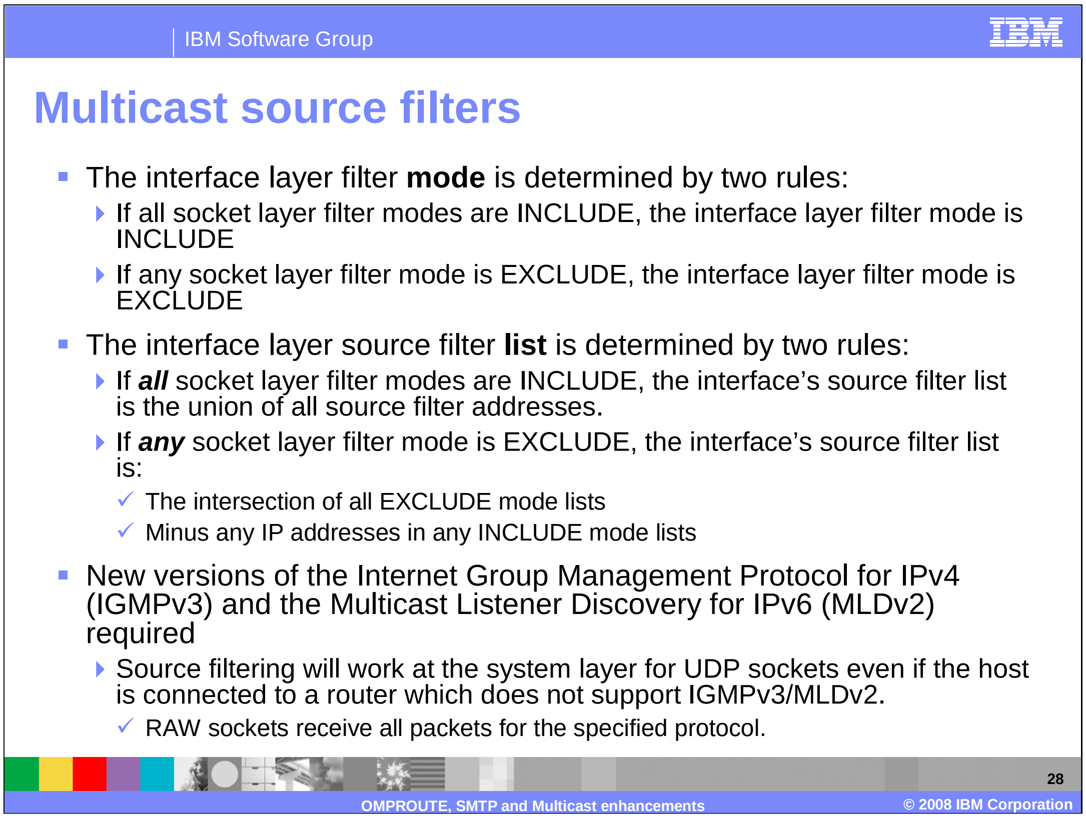
OMPROUTE, SMTP and Multicast enhancements

© 2008 IBM Corporation

New API's are required to allow the client programs to specify a source filter list and what is referred to as a filter mode. There are two types of API's for multicast source filtering. Delta based or Basic adds or deletes to the source list, can have only one entry on a single call and does not allow changing the filter mode. Full state or Advanced allows the full replacement of a source filter list and the filter mode on a single call. This support is available on the following APIs: z/OS Language Environment C/C++, Macro – EZASMI, z/OS UNIX System Services: Assembler Callable Services, Callable – EZASOKET, CICS, and REXX. Refer to the appropriate documentation for more details about Multicast Source Filter APIs.

The filter mode can be either INCLUDE or EXCLUDE. With INCLUDE mode, a client specifies which multicast datagram's they want to receive, based on the source IP address. With EXCLUDE mode, a client specifies which multicast datagram's they don't want to receive, based on the source IP address.

The source filter mode and a source filter list are maintained at both the socket layer and the interface layer. At the socket layer, the source filter mode and the source list reflect what was specified by the application. The filter mode and the source filter list for the interface is derived from all socket layer filter modes and source filter lists which have joined a multicast group for the interface. Information at the interface layer is what is reported to multicast routers. This allows the multicast routers to determine which multicast datagrams to forward.



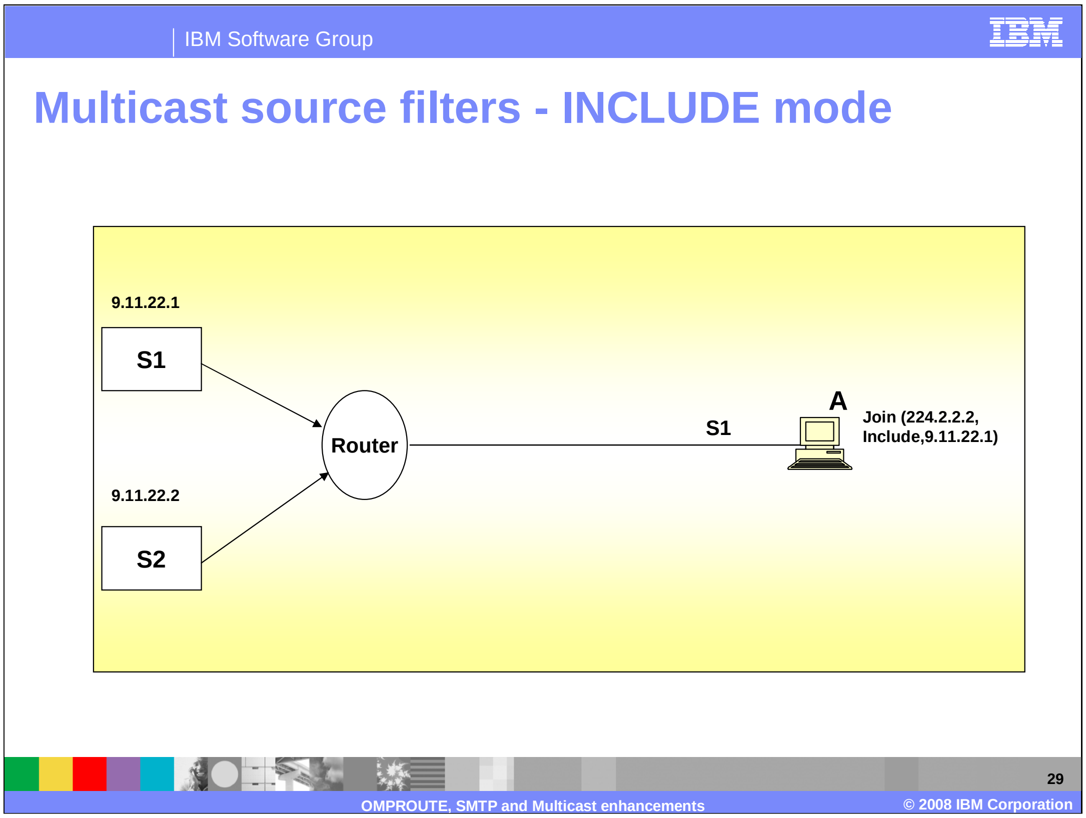
Multicast source filters

- The interface layer filter **mode** is determined by two rules:
  - If all socket layer filter modes are INCLUDE, the interface layer filter mode is INCLUDE
  - If any socket layer filter mode is EXCLUDE, the interface layer filter mode is EXCLUDE
- The interface layer source filter **list** is determined by two rules:
  - If ***all*** socket layer filter modes are INCLUDE, the interface's source filter list is the union of all source filter addresses.
  - If ***any*** socket layer filter mode is EXCLUDE, the interface's source filter list is:
    - ✓ The intersection of all EXCLUDE mode lists
    - ✓ Minus any IP addresses in any INCLUDE mode lists
- New versions of the Internet Group Management Protocol for IPv4 (IGMPv3) and the Multicast Listener Discovery for IPv6 (MLDv2) required
  - Source filtering will work at the system layer for UDP sockets even if the host is connected to a router which does not support IGMPv3/MLDv2.
    - ✓ RAW sockets receive all packets for the specified protocol.

The interface layer filter mode is determined by two rules; 1) If all socket layer filter modes are INCLUDE, the interface layer filter mode is INCLUDE. 2) If any socket layer filter mode is EXCLUDE, the interface layer filter mode is EXCLUDE.

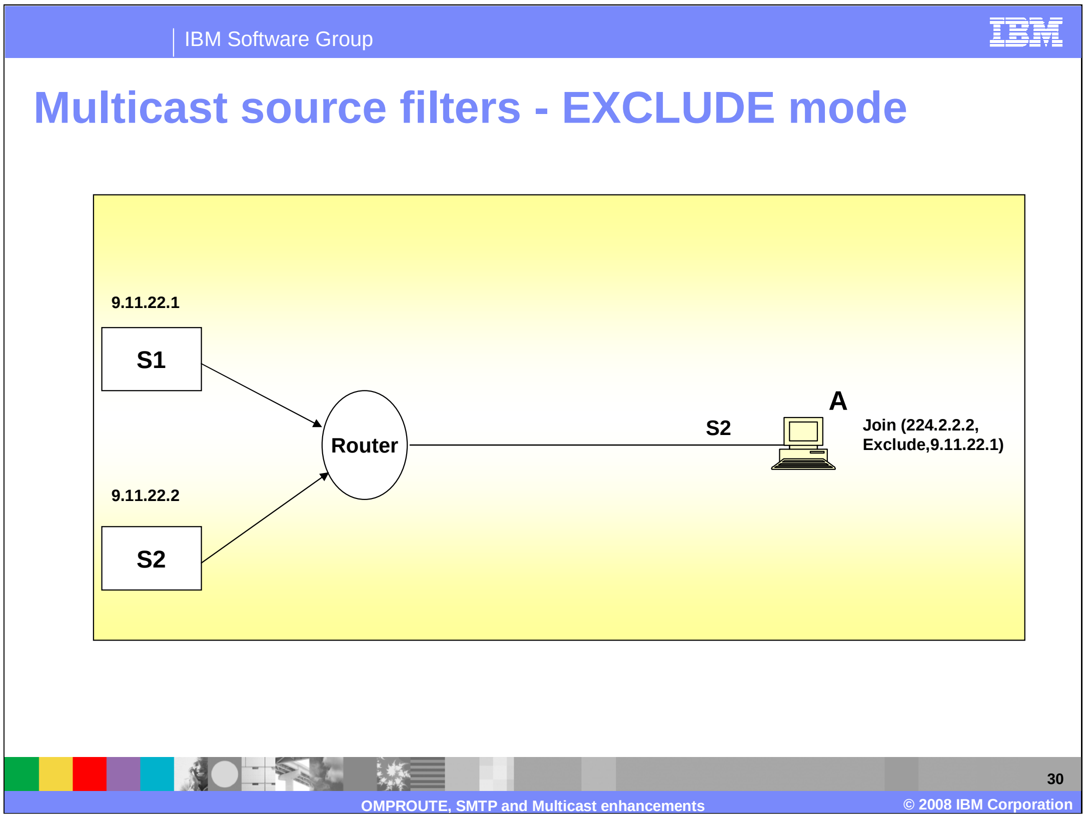
The interface layer source filter list is determined by two rules; 1) If all socket layer filter modes are INCLUDE, the interface's source filter list is the union of all source filter addresses. 2) If any socket layer filter mode is EXCLUDE, the interface's source filter list is derived by taking the intersection of all EXCLUDE mode lists minus any IP addresses in any INCLUDE mode lists.

New versions of IGMP and MLD are required to communicate the new filtering information to multicast routers. Note that this solution also allows the local system to filter on source addresses even if the system is not attached to a multicast router which supports source address filtering.
RFC3678 defines new socket options and functions to manage source filters. RFC 3376 and 3810 define the IGMPv3 (IPv4) and MLDv2 (IPv6) protocols used by systems to report their IP multicast group memberships to neighboring multicast routers. With the new versions of the protocols, multicast routers are informed of the source IP filtering of any applications on a system. This allows the multicast router to send only multicast datagrams, which the system has applications interested in receiving. z/OS communications Server does not support any multicast routing protocols and only supports source filtering for user datagram protocol (UDP) sockets.



Multicast source filters - INCLUDE mode

In this example, client A specifies it wants to receive multicast datagram's from server 1 only. All multicast datagram's from server 2 are dropped by the router.



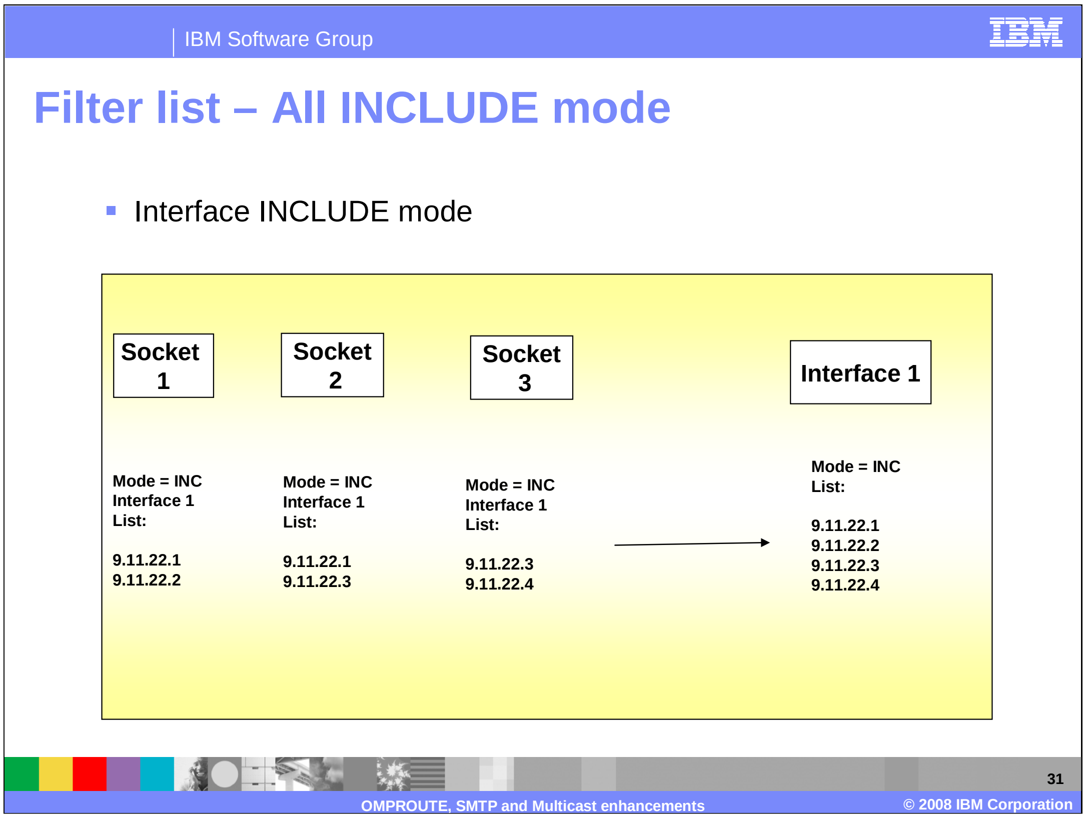
IBM Software Group

Multicast source filters - EXCLUDE mode

30

OMPROUTE, SMTP and Multicast enhancements

In this example, client A specifies it doesn’t want to receive multicast datagram’s from server 1. All multicast datagram’s from server 1 are dropped by the router.



IBM Software Group

Filter list – All INCLUDE mode

- Interface INCLUDE mode

Socket 1

Mode = INC
Interface 1
List:

9.11.22.1
9.11.22.2

Socket 2

Mode = INC
Interface 1
List:

9.11.22.1
9.11.22.3

Socket 3

Mode = INC
Interface 1
List:

9.11.22.3
9.11.22.4

Interface 1

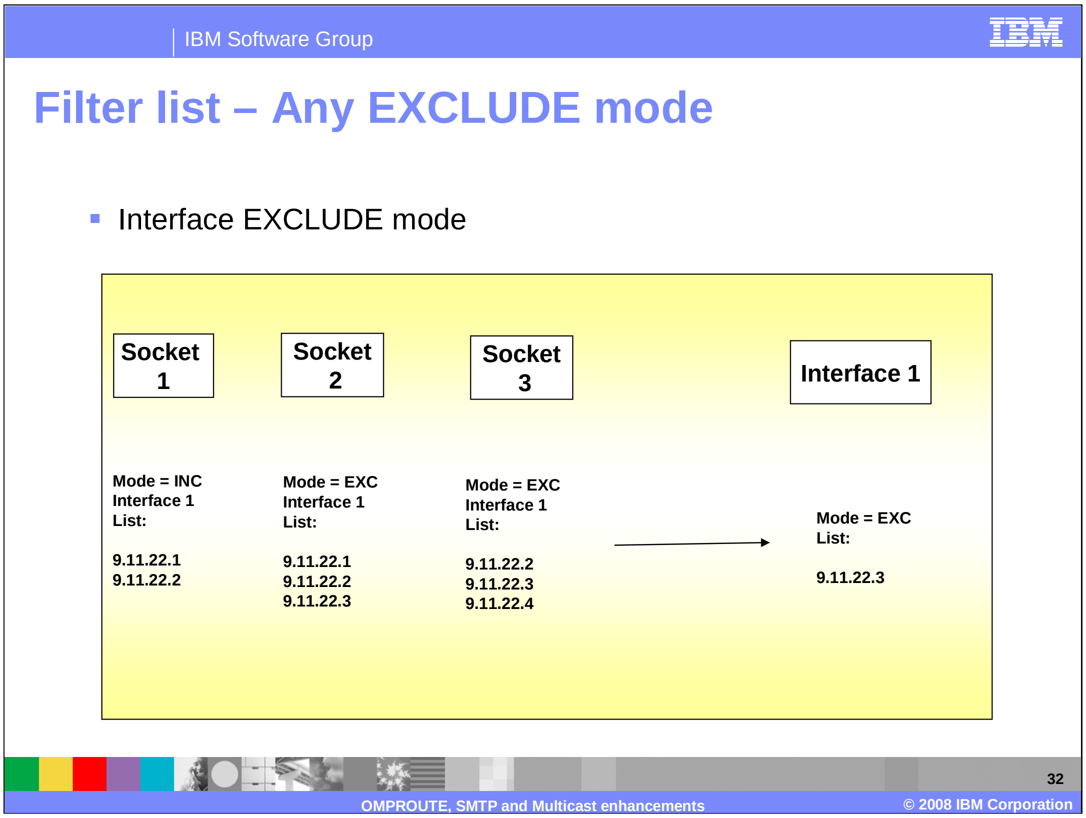
Mode = INC
List:

9.11.22.1
9.11.22.2
9.11.22.3
9.11.22.4

31

OMPROUTE, SMTP and Multicast enhancements

In this example, all three sockets have specified INCLUDE mode. Therefore the interface layer's filter mode is also INCLUDE and it's source filter list is the union of the socket layer's source filter list.



IBM Software Group

Filter list – Any EXCLUDE mode

- Interface EXCLUDE mode

Socket 1

Mode = INC
Interface 1
List:

9.11.22.1
9.11.22.2

Socket 2

Mode = EXC
Interface 1
List:

9.11.22.1
9.11.22.2
9.11.22.3

Socket 3

Mode = EXC
Interface 1
List:

9.11.22.2
9.11.22.3
9.11.22.4

Interface 1

Mode = EXC
List:

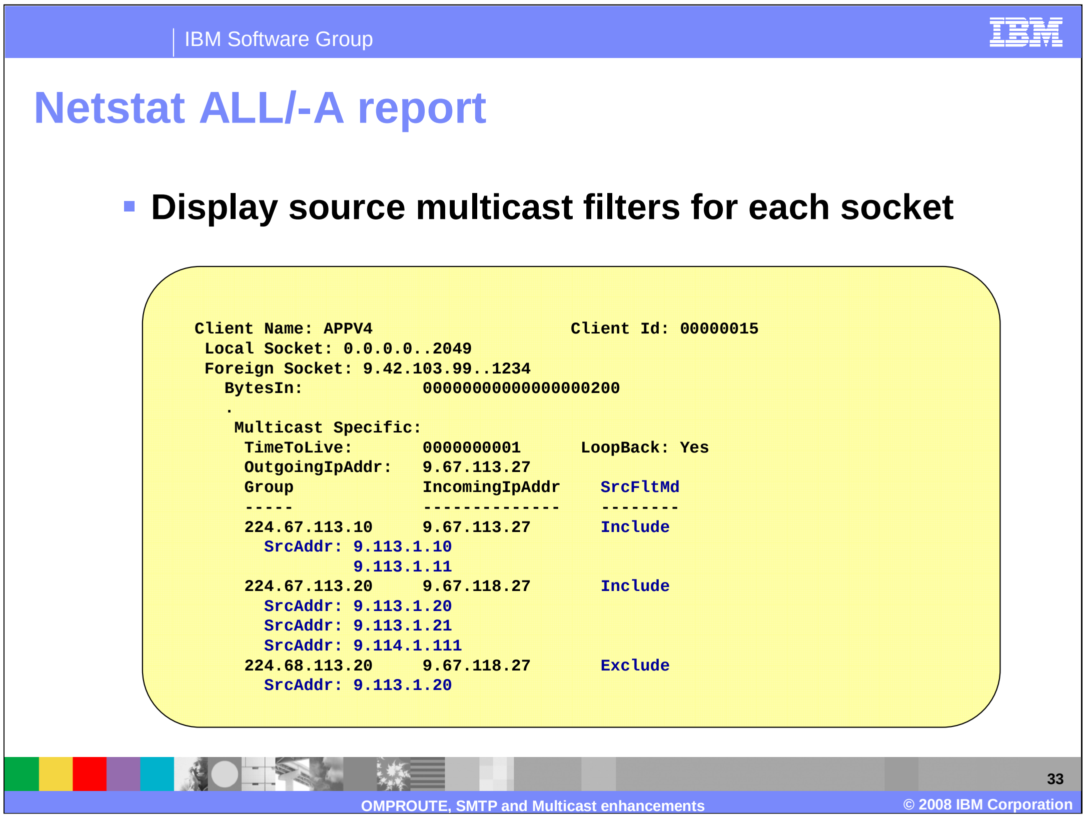
9.11.22.3

32

OMPROUTE, SMTP and Multicast enhancements

© 2008 IBM Corporation

In this example, socket 1 specified a filter mode of INCLUDE and sockets 2 and 3 specified a filter mode of EXCLUDE. The interface layer mode is therefore EXCLUDE. The source filter list is derived by first taking the intersection of socket 2 and socket 3's lists. In this case the intersection is 9.11.22.2 and 9.11.22.3. Then any addresses which are in socket 1's source list are deleted. In this case 9.11.22.2 is in the list so it is deleted from the interface list. The interface layer therefore has a filter mode of EXCLUDE and a source filter list of 9.11.22.3.



IBM Software Group

Netstat ALL/-A report

- Display source multicast filters for each socket

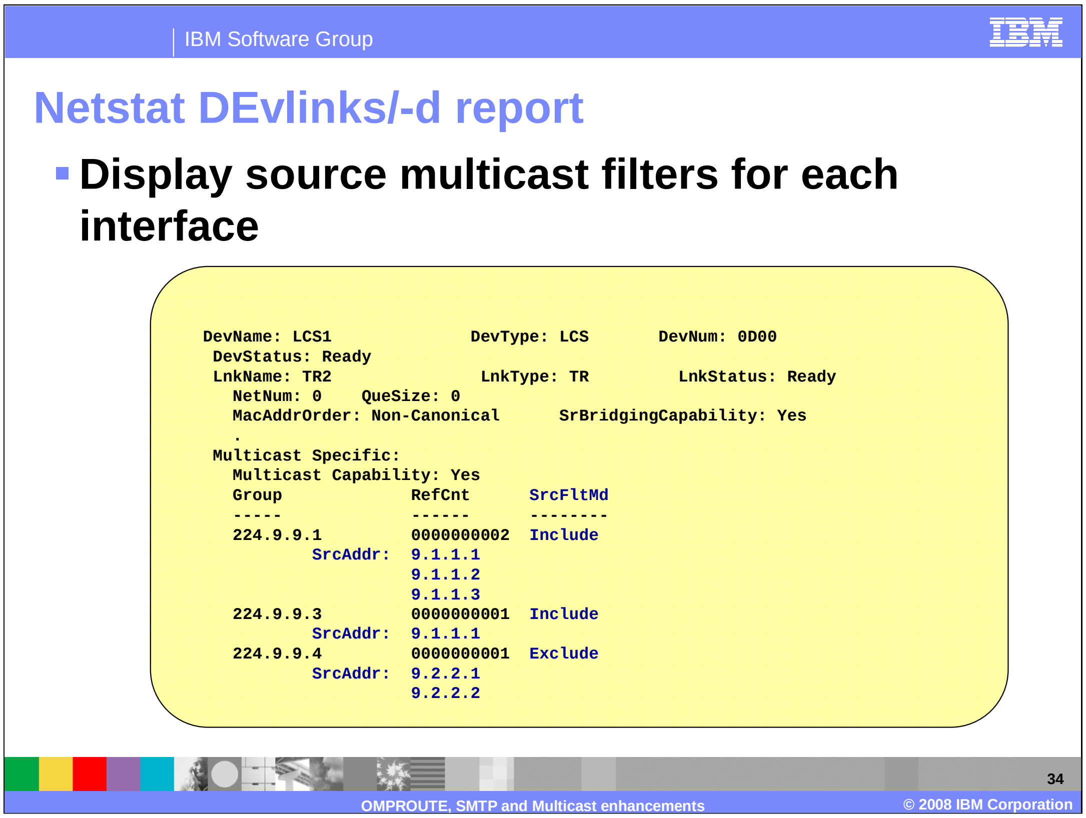
```
Client Name: APPV4                    Client Id: 00000015
 Local Socket: 0.0.0.0..2049
 Foreign Socket: 9.42.103.99..1234
   BytesIn:            00000000000000000200
   .
    Multicast Specific:
     TimeToLive:       0000000001      LoopBack: Yes
     OutgoingIpAddr:   9.67.113.27
     Group             IncomingIpAddr    SrcFltMd
     -----             --------------    --------
     224.67.113.10     9.67.113.27       Include
       SrcAddr: 9.113.1.10
                9.113.1.11
     224.67.113.20     9.67.118.27       Include
       SrcAddr: 9.113.1.20
       SrcAddr: 9.113.1.21
       SrcAddr: 9.114.1.111
     224.68.113.20     9.67.118.27       Exclude
       SrcAddr: 9.113.1.20
```

33

OMPROUTE, SMTP and Multicast enhancements

© 2008 IBM Corporation

The netstat ALL/-A report has been updated to include the source filter mode and source filter list.

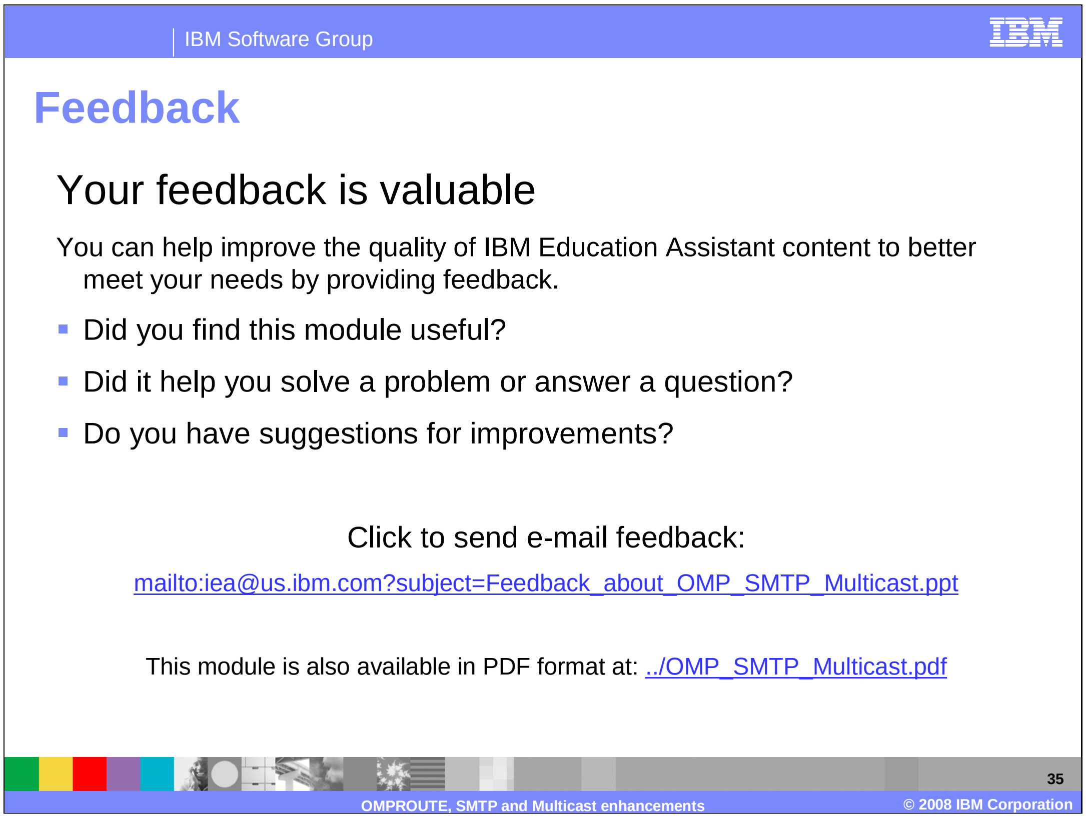


Netstat DEvlinks/-d report

- **Display source multicast filters for each interface**

```
DevName: LCS1              DevType: LCS       DevNum: 0D00
 DevStatus: Ready
 LnkName: TR2               LnkType: TR         LnkStatus: Ready
   NetNum: 0    QueSize: 0
   MacAddrOrder: Non-Canonical      SrBridgingCapability: Yes
   .
 Multicast Specific:
   Multicast Capability: Yes
   Group             RefCnt      SrcFltMd
   -----             ------      --------
   224.9.9.1         0000000002  Include
           SrcAddr:  9.1.1.1
                     9.1.1.2
                     9.1.1.3
   224.9.9.3         0000000001  Include
           SrcAddr:  9.1.1.1
   224.9.9.4         0000000001  Exclude
           SrcAddr:  9.2.2.1
                     9.2.2.2
```

The netstat DEVLINKS/-d report has been updated to display the source filter mode and the source filter lists.



Feedback

Your feedback is valuable

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

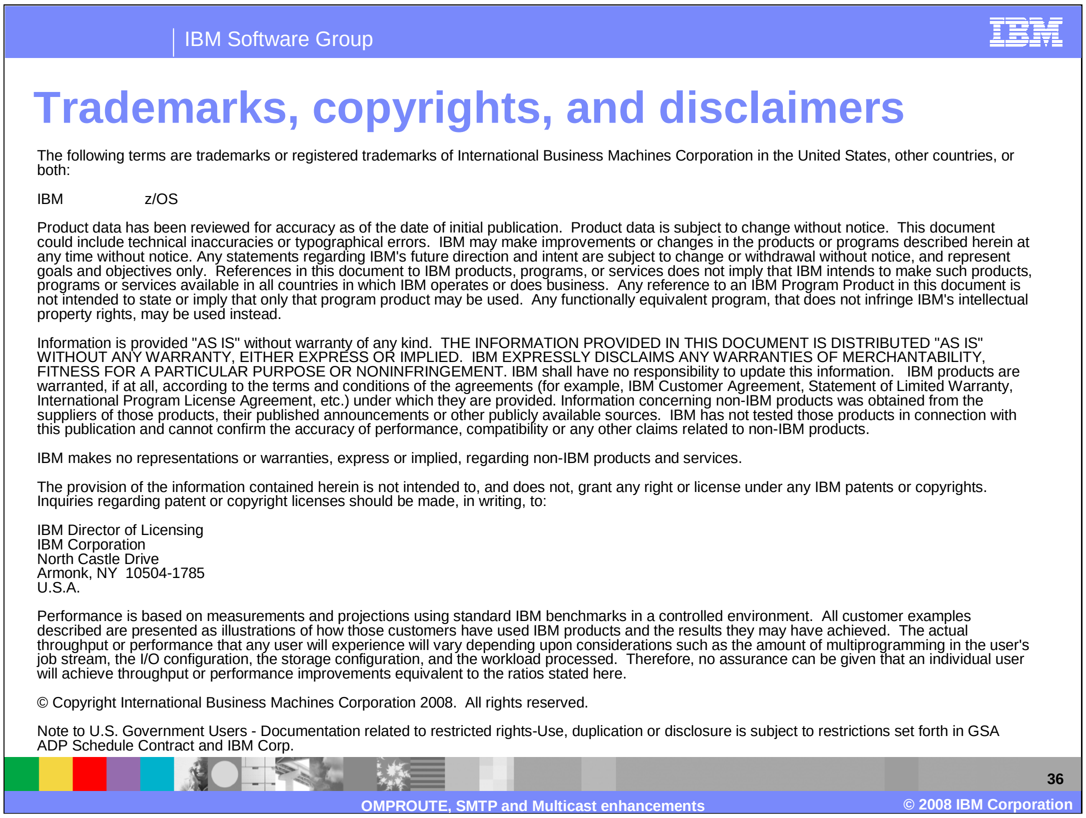
- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

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

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

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



Trademarks, copyrights, and disclaimers

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

IBM z/OS

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

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

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

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

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

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

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