IBM Software Group

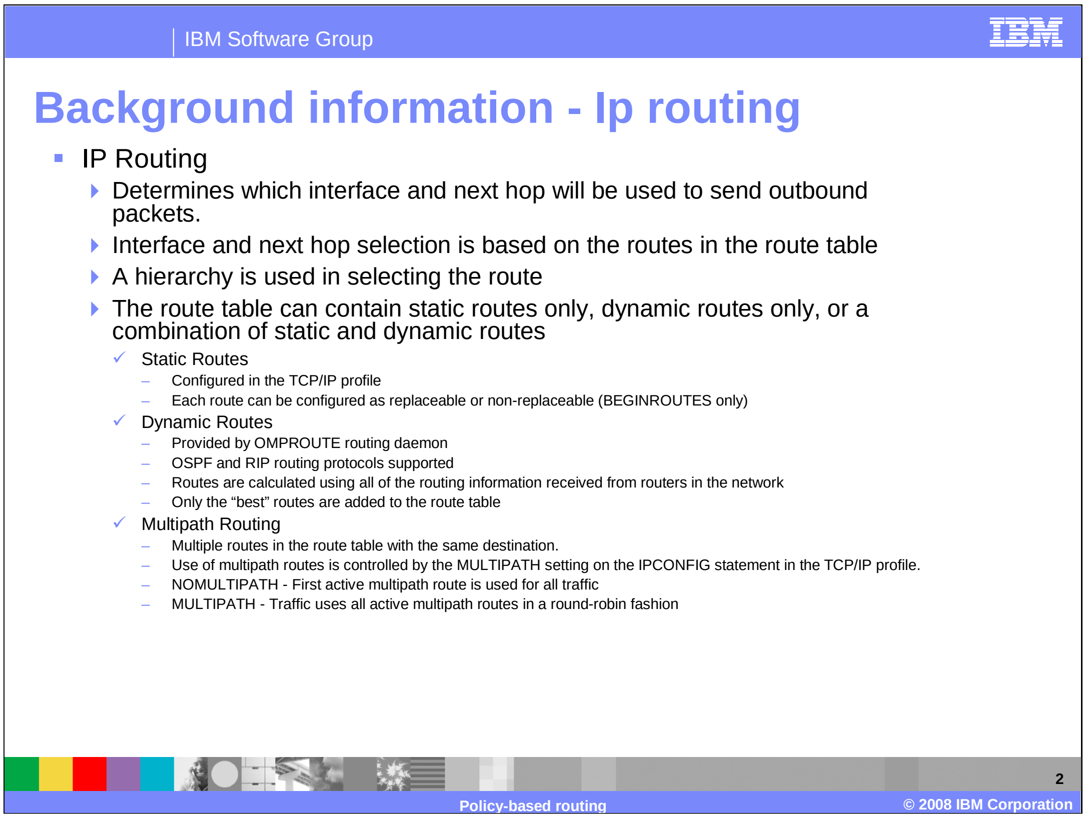
z/OS V1R9 Communications Server

Policy-based routing

business on demand.

Updated January 11, 2008

This presentation discusses the Policy-based Routing enhancements in z/OS V1R9 Communications Server.



IP Routing is the TCP/IP stack function that uses a table of routes to determine which interface and next hop will be used for IP traffic that is leaving the stack. The IP route table may contain only static routes, or it may contain only dynamic routes, or it may contain a combination of the two. It also may contain a combination of routes for single destinations, known as host routes, routes for all destinations in a IP subnet, IP network, or IP supernet, and routes that can be used for any destination, known as default routes. When IP Routing is searching the route table for a route to be used for sending traffic to a destination, it searches for the first active route that includes the destination IP address, in the order host, subnet, network, supernet, default.

Static routes in the z/OS Communications Server route table are configured in the TCP/IP profile using either the BEGINROUTES statement or the GATEWAY statement. The BEGINROUTES statement is an alternative to the GATEWAY statement that allows addresses to be specified using a BSD style syntax and that has some enhancements that are not available with the GATEWAY statement. When the BEGINROUTES statement is used, each static route can be configured as either replaceable or non-replaceable. This setting determines whether the route can be replaced by dynamic routes that are learned by the OMPROUTE routing daemon. Replaceable routes can be replaced by dynamic routes learned by OMPROUTE. Non-replaceable routes cannot be replaced by dynamic routes learned by OMPROUTE.

Dynamic routes in the z/OS Communications Server route table are provided by the routing daemon OMPROUTE. OMPROUTE uses information learned from routers in the network, via either the OSPF or RIP routing protocol, to calculate the dynamic routes. This information may provide many different routes to network destinations, but only the “best” routes to each destination are added to the stack route table. The “best” routes are determined by assigning cost values to each route, based on configuration information within OMPROUTE and the network routers.

Multipath is a function of IP Routing that determines the processing to be performed when there are multiple routes in the route table to the same destination. There can be multiple dynamic routes added to the same destination, as mentioned on the previous slide, or you can configure multiple static routes to the same destination. All static routes to a destination are considered to have the same cost and all are added to the route table. When the routing information learned by OMPROUTE results in multiple dynamic routes to a destination with the same cost, all are added to the route table. The Multipath function is controlled by a setting on the IPCONFIG statement in the TCP/IP profile. When NOMULTIPATH is specified on the IPCONFIG statement and multiple routes to a destination are in the route table, all traffic sent to that destination uses the first active route to the destination. The other routes to the destination provide backup should the first route become inactive, but they are not used as long as the first route is active. When MULTIPATH is specified on the IPCONFIG statement and multiple routes to a destination are in the route table, each of the active routes are used in a round-robin fashion. The method in which they are used is controlled by a qualifier configured on the MULTIPATH parameter. The possible values for this qualifier are PERConnection and PERPacket. When PERConnection is specified then for TCP, the next multipath route is selected for each new connection. The connection uses that route as long as the route is usable. For UDP/RAW, the next multipath route is selected for each new route lookup. When PERPacket is specified then the next multipath route is selected for each packet sent.

IBM Software Group

Problem - Limited criteria for route selection

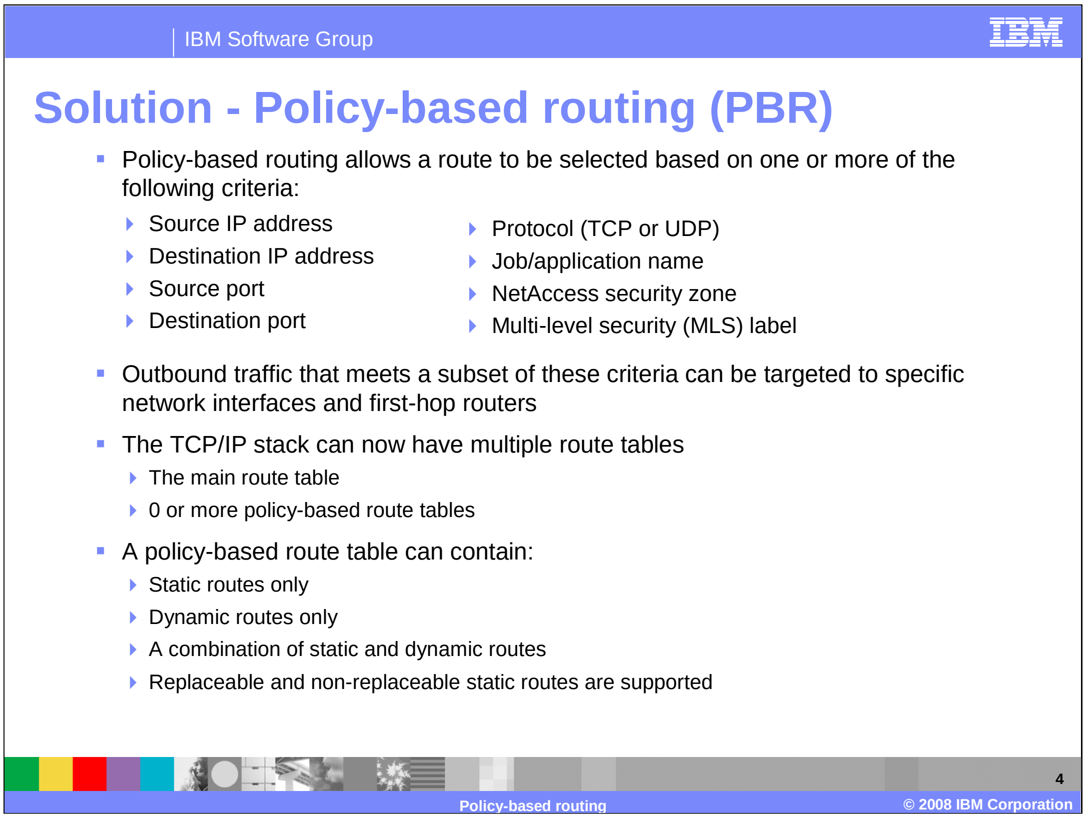
- IP routing selects a route for an outbound packet based solely on the packet’s destination IP address
- All traffic being sent to a destination address must use the same route (or group of multipath routes)
- Traffic being sent to that destination that also meets certain other criteria cannot be made to use a different route

3

Policy-based routing

As covered in the Background section, when IP Routing needs to send outbound traffic, it searches the route table for a route that matches the destination IP address of the traffic. This may be a host route specific to the destination address, or a route to a subnet, network, or supernet containing the destination address, or it may be the default route which covers all destination addresses.

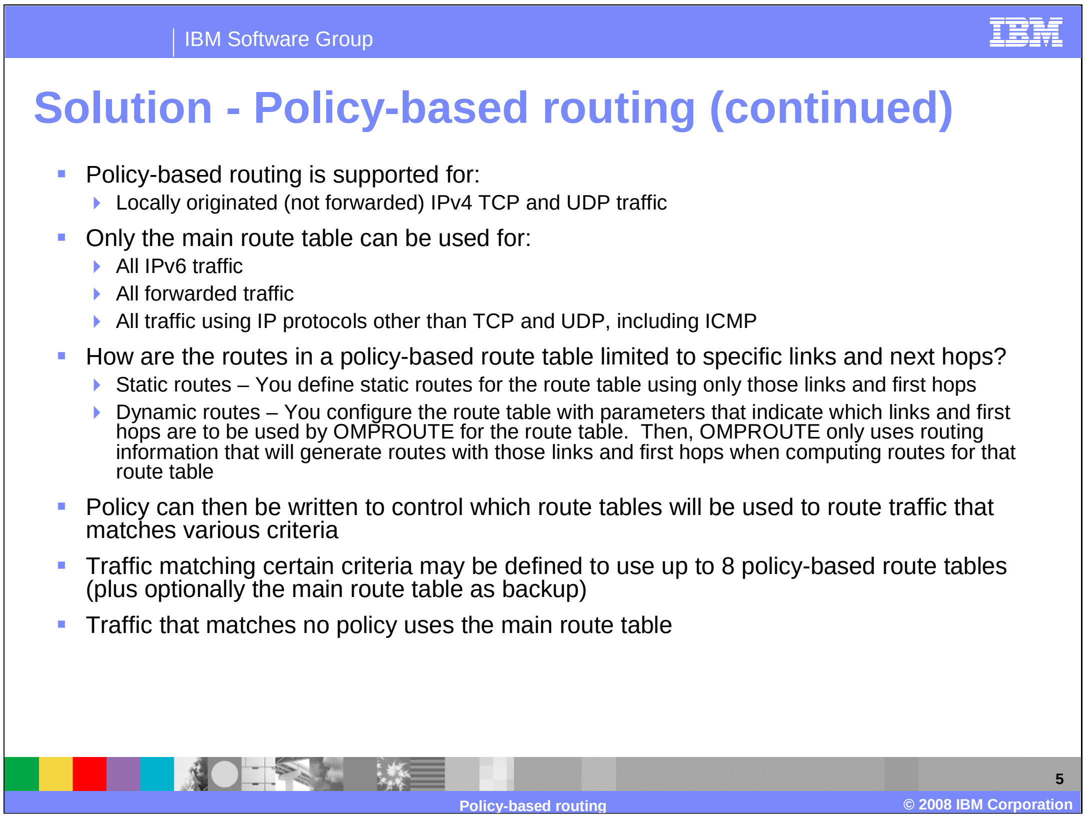
A limitation that has existed with IP Routing is due to the fact that only the destination IP address could be used when selecting a route for outbound traffic. All traffic destined for a particular IP address had to use the same route or group of multipath routes. There has been no way to use different routes for different purposes such as for FTP traffic, for secure traffic, for Enterprise Extender traffic, etc.



Policy-based routing addresses this limitation of IP Routing by allowing a route to be selected based on much more than just the destination IP address. The additional route selectors, which are listed on this slide, can be used to cause traffic that meets more specific criteria to be targeted to specific network interfaces and first-hop routers.

So, how does policy-based routing allow IP Routing to use these additional route selectors?

It is made possible through the use of multiple route tables. In addition to the main route table, the TCP/IP stack can now have multiple policy-based route tables. Policy-based route tables have many of the same characteristics as the main route table. They can contain both static and dynamic routes and the static routes can be configured as both replaceable and non-replaceable.

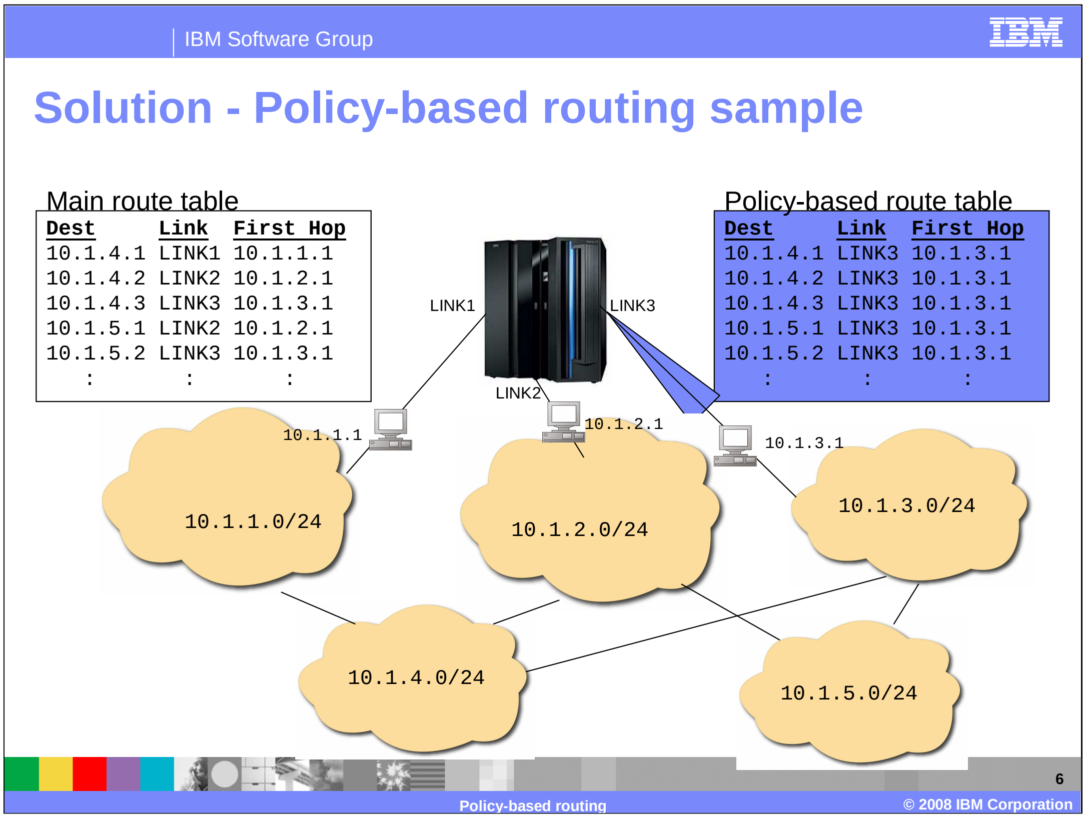


Policy-based routing is not supported for all types of IP traffic. The support is limited to locally originated IPv4 TCP and UDP traffic. All IPv6 traffic, all forwarded traffic, and all traffic using protocols other than TCP and UDP is not processed by policy-based routing and continues to be routed using only the main route table. For example, ICMP Echo request packets sent by the Ping command will continue to be routed using the main route table.

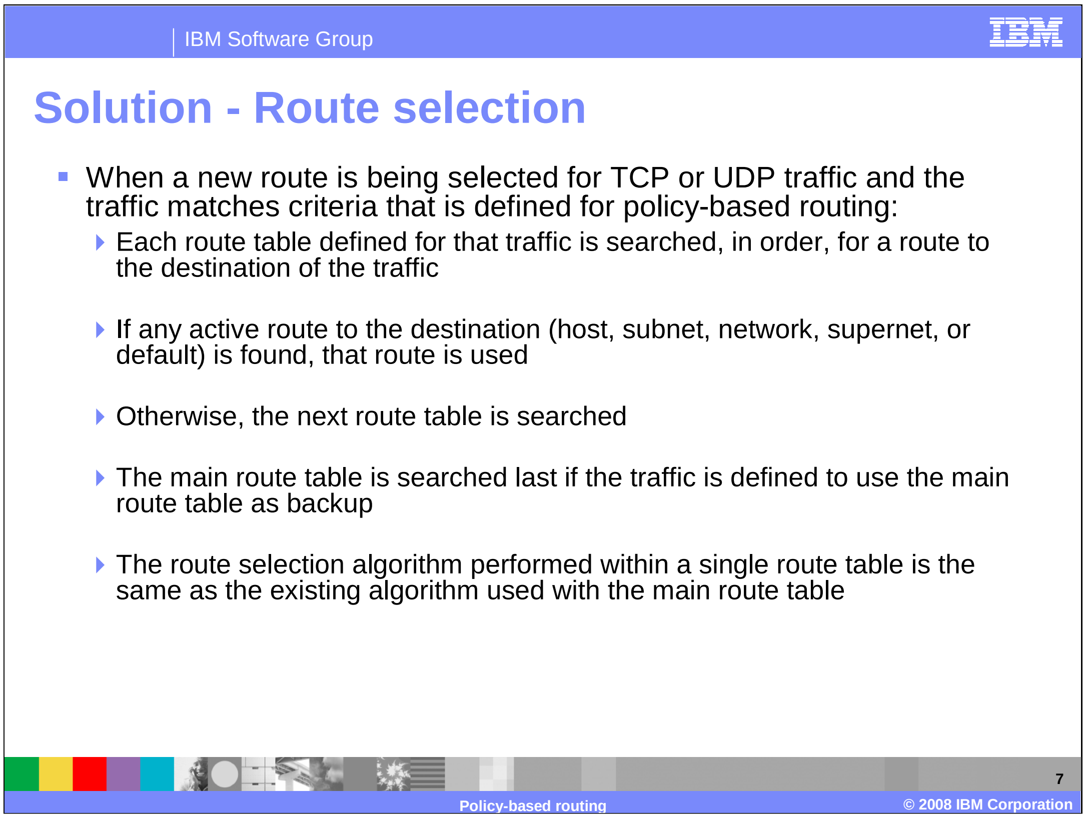
Each policy-based route table can be configured such that all static routes and dynamic routes that it contains are limited to specific links and next hop routers. Static routes are limited to those links and next hops simply by using only those links and next hops on any static routes configured for the policy-based route table. For dynamic routing, where the routes will be added to the route table by OMPROUTE, it is necessary to control the way in which OMPROUTE computes those routes. This is done, for each policy-based route table, through the configuration of dynamic routing parameters which identify the link and next hops that OMPROUTE may use when computing routes for the table.

Once you have policy-based route tables created, how do you cause different types of traffic to use the different route tables?

Using policy agent, policy can be written which indicates that traffic that matches particular combinations of the various route selection criteria will be routed using certain route tables. A particular type of traffic can be defined to use up to 8 policy-based route tables, plus the main route table as backup. Traffic that matches none of the defined policies continues to be routed using the main route table.



In this sample, there is a node connected to a set of IP subnets. You can see, in the partial table shown, that the main route table contains routes to destinations throughout the network and that these routes use all of the three available network links. These may be routes that were added to the main route table by OMPROUTE, in which case the location of the destinations and the dynamic routing configuration throughout the network has resulted in these routes being the best routes available. If there is a need for a certain type of IP traffic (for example all traffic sent by a specific job name) to be sent out LINK3, a policy-based route table such as the one shown could be created. In this particular case, the policy-based route table contains routes to all of the same destinations as are in the main route table. However, all of the routes in the policy-based route table use LINK3.



Once the set of route tables that can be used for some type of outbound traffic has been determined, how does IP Routing search for a route in those tables?

Most often there will be one policy-based route table defined to be used for the traffic, but there may be as many as eight. Each of the policy-based route tables is searched, in the order defined, for a route to the destination. If any active route to the destination is found in a route table, the search is stopped and that route is used for the traffic. This route may be a host route, a subnet, network, or supernet route, or a default route. If no active route to the destination is found in a route table, the search continues with the next route table. If all policy-based route tables are searched without success, the main route table may also be searched if the policy indicates that the main route table can be used as a backup.

Route selection within a route table occurs in the following order:

•If a route exists to the destination address (a host route), it is chosen.

•If no host route exists to the destination address:

  •If subnet, network, or supernet routes exist to the destination, the route with the most specific network mask (the mask with the most bits on) is chosen.

  •If the destination is a multicast destination and a multicast default route exists, that route is chosen.

Default routes are chosen when no other route exists to a destination.

PBR configuration

- Policy-based routing (PBR) is configured in a policy agent flat-file
  - Consists of routing rules, routing actions, and route tables
  - RoutingRule
    - Specifies a set of traffic characteristics and the RoutingAction to be taken for outbound traffic that matches those characteristics. It consist of:
      - Source IP address
      - Destination IP address
      - Traffic descriptor – traffic characteristics
      - Priority
      - Time condition
      - Reference to a RoutingAction
  - RoutingAction
    - Indicates the route tables to be used for traffic that matches a referencing RoutingRule
      - Identifies up to 8 policy-based route tables
      - Route tables are searched in the order specified
      - UseMainRouteTable
  - RouteTable
    - Defines a policy-based route table. It consists of:
      - Table name
      - Route entries - static routes
      - DynamicRoutingParms entries - control calculation of dynamic routes by OMPROUTE
      - Advanced parameters
- No LDAP file support for PBR
- Centralized policy support for PBR
- IBM Configuration Assistant for z/OS Communications Server (Configuration Assistant)
  - Can be used to generate a PBR configuration flat-file

Policy-based routing is configured in a policy agent flat file and it is supported by the centralized policy function. You can manually create the policy agent flat-file or you can use the IBM Configuration Assistant for z/OS Communications Server to generate the file. The items configured for policy-based routing consist of routing rules, routing actions, and route tables.

The RoutingRule statement is where you will identify a type of traffic that you want to be routed using policy-based routing. The traffic can be identified by any combination of its source IP address, its destination IP address, and any of a set of other traffic characteristics in the traffic descriptor. Each RoutingRule can be given a priority and a time condition. Finally, each RoutingRule will reference a RoutingAction which will define the action to be taken for the traffic. The source and destination IP address, if specified in a RoutingRule, indicate the source or destination IP address used in the type of outbound traffic being defined. The source IP address for an outbound TCP connection or an outbound UDP packet can be influenced by several configuration and application options. See the source IP address information in *z/OS Communications Server: IP Configuration Guide* for the hierarchy of ways that the source IP address of an outbound packet is determined.

The traffic descriptor defines the remainder of the characteristics that can be used to identify a type of traffic that will use policy-based routing. If the traffic descriptor is used to identify a particular type of traffic, it can be configured inline in the RoutingRule or the RoutingRule can reference one or more previously configured traffic descriptors. A traffic descriptor can specify any combination of source and destination port, traffic protocol, job name of the sending application, NetAccess security zone of the traffic, and MLS security label of the traffic. An outbound packet's destination IP address is used to determine the packet's NetAccess security zone in the NetAccess table defined in the TCP/IP profile. The MLS security label is the label associated with the NetAccess security zone. Each RoutingRule can be configured with a priority value, which is used to select a rule for outbound traffic when the traffic could match the characteristics configured for multiple rules. If these rules are not configured with different priority values, the precedence of the rules is unpredictable. Note that rule priority is not explicitly configured when the IBM Configuration Assistant for z/OS Communications Server is used to configure policy-based routing. In that case, rule priority is determined by the order of the rules as shown on the rules panel. The time condition of a RoutingRule controls when the rule is active and installed in the TCP/IP stack. The reference to a RoutingAction provides a link to the RoutingAction that will define the action to be taken for the type of traffic defined by the RoutingRule.

The RoutingAction statement is where you will identify the set of policy-based route tables that will be used to route traffic that you have defined with a RoutingRule statement. A RoutingAction can specify up to eight policy-based route tables that will be searched, in order, to find a route to the destination of the traffic. In addition, the RoutingAction may indicate that the main route table should also be searched when a usable route is not found in any of the policy-based route tables specified.

The RouteTable statement is where you will define the characteristics of each policy-based route table. To define a policy-based route table, you will need to provide a name for the table. In addition, you may define static routes to be added to the table, dynamic routing parameters to control the dynamic routes that will added to the table by OMPROUTE, as well as a few advanced table parameters.

More on policy-based route tables

- Up to 255 policy-based route tables can be defined for a TCP/IP stack
- Only active route tables are installed in the TCP/IP stack
- A policy-based route table can contain static routes, dynamic routes, or both (IPv4 routes only)
- The Route table name uniquely identifies a policy-based route table
  - 1-8 characters long
  - The names EZBMAIN and ALL (in lower, upper, and mixed case) are reserved
- A Route entry defines a static route
  - Syntax similar to that of BEGINROUTES. Differences shown below:

	RouteTable Route entry	BEGINROUTES
Destination - Single IP address	Ipaddress Or Ipaddress/32	ipaddress HOST Or Ipaddress/32
Destination - Range of IP addresses	ipaddress/prefixLength	ipaddress address_mask Or Ipaddress/num_mask_bits
Link name undefined in TCP/IP profile	Route created but not usable until link is defined. Netstat shows status of "I"	Route rejected by profile processing.

Up to 255 policy-based route tables can be defined for a stack, but only the active tables are installed in the stack. A route table is active if it is referenced by an active routing rule and its associated action. Like the main route table, a policy-based route table can contain both static and dynamic routes.

The table name specified for a policy-based route table may be from one to eight characters in length. With the addition of the policy-based routing function, the main route table has also been given a name so it can be identified in displays and messages. The name of the main route table is EZBMAIN.

Policy-based route table names can be configured in lower case, upper case, or mixed case. Reports presented on the MVS console (for example, Netstat and OMPROUTE Display) are displayed in all upper case. When a name is provided to filter Netstat or OMPROUTE Display output, the case of the name is ignored. If the same name is used for multiple policy-based route tables, but using a different case for each the names will be indistinguishable in MVS console reports and when reports are filtered by table name, all tables with that name will be included, regardless of case. You may want to define all table names using UPPER case.

A Route entry on the RouteTable statement is used to define a static route. The syntax of the Route entry is similar to the syntax of the ROUTE entry on the BEGINROUTES statement, used to define static routes for the main route table. The differences between the two are in the way that the route destination is specified and in the way that a route is processed by the stack when the link that the route uses is not defined to the stack.

More on policy-based route tables (continued)

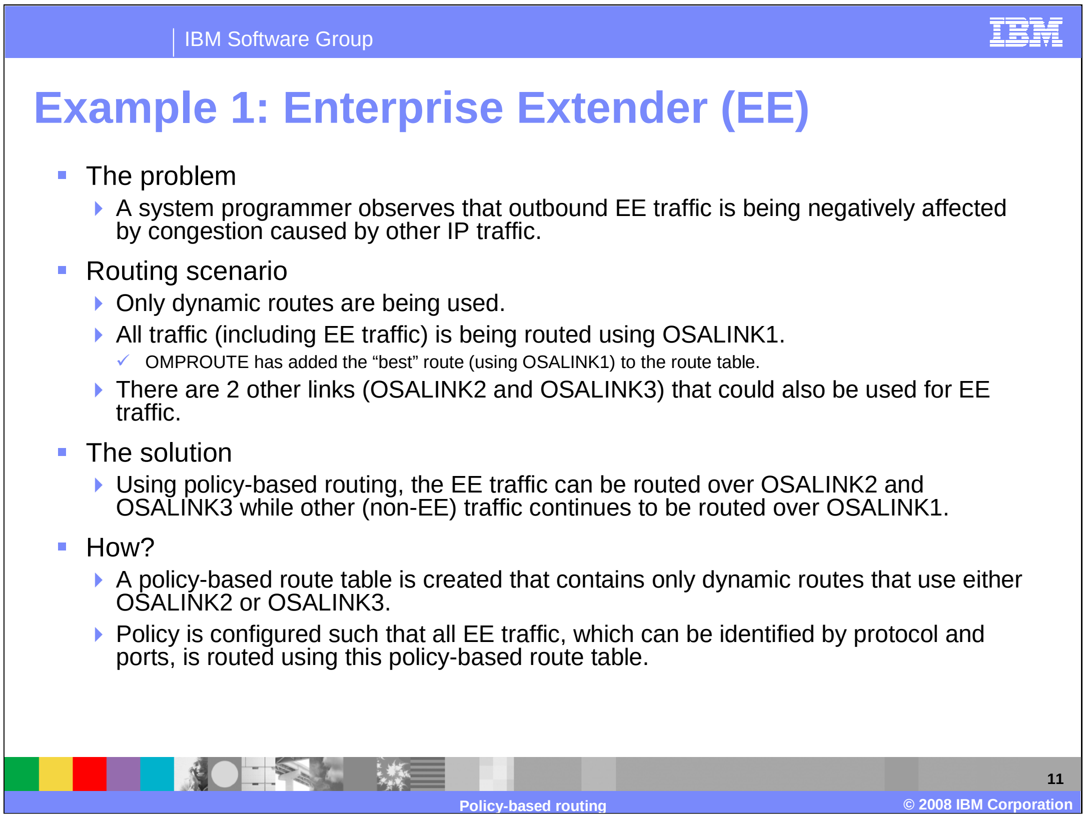
- A DynamicRoutingParms entry defines parameters used by OMPROUTE to control the dynamic routes added to the policy-based route table
  - Multiple DynamicRoutingParms can be configured on a RouteTable statement
- Route Table Advanced Parameters
  - Multipath indicates whether the Multipath algorithm should be used for this table
  - IgnorePathMtuUpdate indicates whether IPv4 ICMP Fragmentation Needed messages should be ignored for this route table
  - DynamicXCFRoutes indicates whether direct routes to dynamic XCF addresses on other TCP/IP stacks should be added to this route table.

A DynamicRoutingParms entry on the RouteTable statement is used to define a link and, optionally, a next hop IP address that are to be used by OMPROUTE to control the dynamic routes that will be added to the route table. If the link is not defined in the TCP/IP profile, the DynamicRoutingParms definition is kept, but not used until the link is defined. Multiple DynamicRoutingParms entries can be specified on a RouteTable statement to allow the route table to use multiple links and next hops.

The Multipath setting on the RouteTable statement allows you to indicate when the multipath algorithm used for a policy-based route table should be different from the algorithm being used for the main route table. The main route table uses the IPCONFIG MULTIPATH / NOMULTIPATH setting from the TCP/IP profile. If a different multipath setting is needed for traffic using a policy-based route table, use the RouteTable Multipath parameter. You can specify UseGlobal which indicates the IPCONFIG multipath setting will be used for this policy-based routing table. You can indicate that either the perpacket or the perconnection multipath algorithm is to be used for the table or you can also indicate that no multipath processing should be used for the table.

The IgnorePathMtuUpdate option allows you to control whether ICMP Fragmentation Needed messages will be applied to the routes in the table. This is an advanced option that should not normally need to be set. When path MTU discovery is enabled for the stack, IPv4 ICMP Fragmentation Needed messages are used to lower the MTU value used to send data to a specific destination. The path MTU is updated for all routes to the destination. By default, all routes to the destination in policy-based route tables are also updated. You may want to ignore path MTU updates for a policy-based route table containing routes known to use paths that support large MTU values. If there are routes in another route table for the same destinations and those routes may require a smaller path MTU value, IgnorePathMtuUpdate Yes will ensure that a path MTU update that results from sending data on a small MTU route will not cause an update to the path MTU for the routes in the policy-based route table.

The DynamicXCFRoutes option allows you to control whether direct routes to dynamic XCF addresses on other TCP/IP stacks should be added to the route table. This is an advanced option that should not normally need to be set. Consider this option if you have locally originated traffic that will use a policy-based route table whose destination will be the dynamic XCF address of another stack. The routes that will be added to the policy-based route table as a result of this option being set are the same routes that are automatically generated in the main route table when dynamic XCF links are active.



As an example of a situation where policy-based routing may be useful, consider the scenario where a system programmer has determined that his outbound Enterprise Extender traffic is being negatively affected by congestion on the link being used by the main route table. His stack is configured to use dynamic routing and the dynamic routes are sending all traffic over the same link. He knows that if he could somehow move only the Enterprise Extender traffic to another available link, he could get that traffic flowing better.

Using policy-based routing he can move the EE traffic. He can create a policy-based route table that will only contain routes that use the other available links. He can then define policy such that all of the EE traffic will use that route table. All of the EE traffic will now be routed using the other available links while all other traffic, which does not match the policy, will continue to be routed using the main route table.

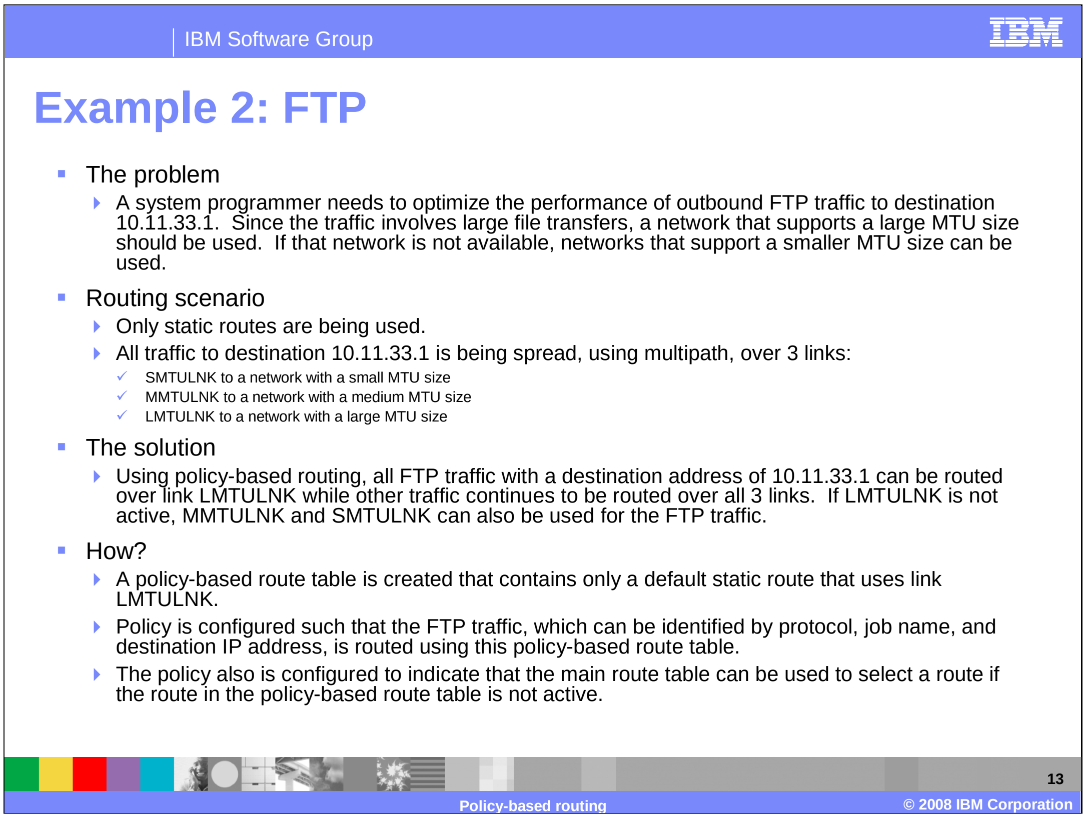
Sample syntax for example 1: EE

```
  RoutingRule EERoutingRule
  {
    TrafficDescriptor
    {
      Protocol              UDP
      SourcePortRange       12000 12004
      DestinationPortRange  12000 12004
    }
    RoutingActionRef  EERoutingAction
  }

RoutingAction EERoutingAction
  {
    UseMainRouteTable No
    RouteTableRef  EERtTbl
  }

 RouteTable EERtTbl
  {
    DynamicRoutingParms OSALINK2 10.11.12.1
    DynamicRoutingParms OSALINK3 10.11.13.1
  }
```


This notes page shows an example of policy that could be written to solve the example problem, where the system programmer needed to move EE traffic from a congested link to other available links.



As another example of a situation where policy-based routing may be useful, consider the scenario where a system programmer needs to optimize the performance of her outbound FTP traffic to a particular destination. Her stack is configured to use static routing and the static routes are configured such that all traffic is being spread across three different links. These links access networks with a variety of MTU sizes. She knows that she could improve the performance of this FTP traffic if she could make all of that traffic go out over the link to the network with the largest MTU, whenever possible.

Using policy-based routing she can move this specific FTP traffic. She can create a policy-based route table that contains only routes that use the link to the network with the largest MTU. She can then define policy such that all of the FTP traffic to the particular destination will use that route table. All of that FTP traffic will now be routed using the large MTU network while all other traffic, which does not match the policy, will continue to be routed using the main route table. Since she wants this FTP traffic to continue to be routed, even if the link to the large MTU network becomes unavailable, she can indicate that the main route table be used as a backup to the policy-based route table.

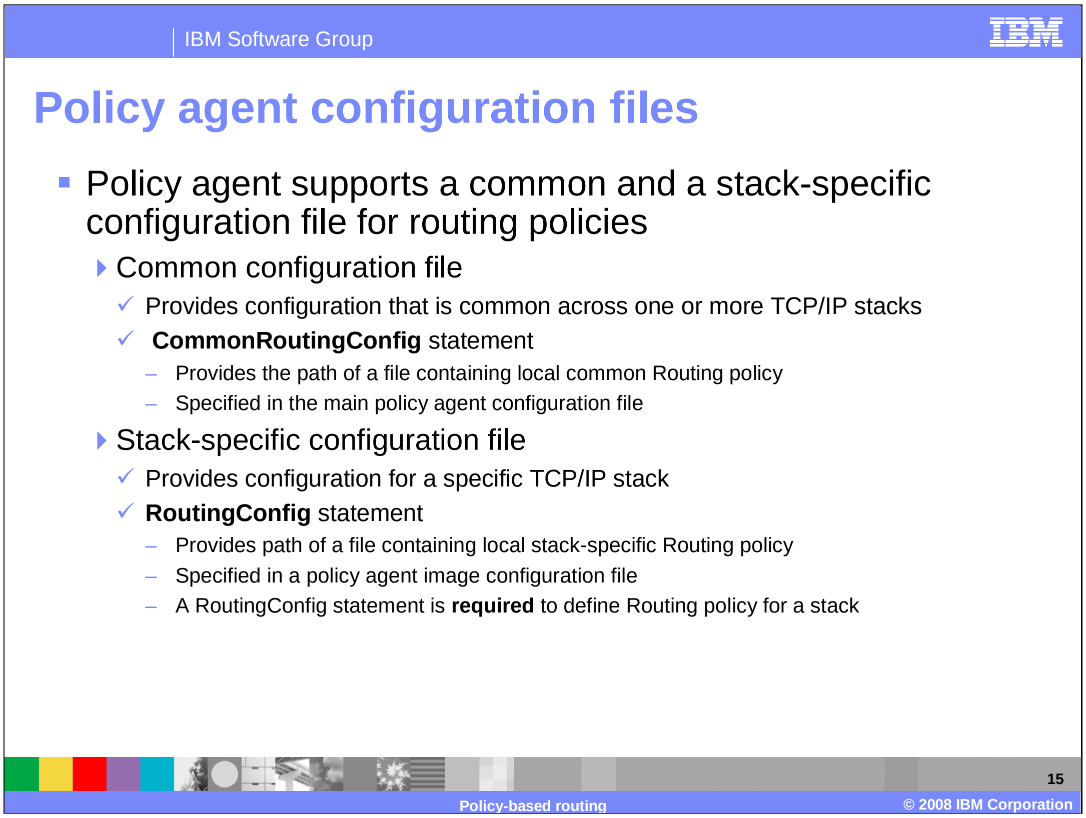
Sample syntax for example 2: FTP

```
RoutingRule FTPRoutingRule
{
  IpDestAddr              10.11.33.1
  TrafficDescriptor
  {
    Protocol              TCP
    JobName               FTP*
  }
  RoutingActionRef  FTPRoutingAction
}

RoutingAction FTPRoutingAction
{
  UseMainRouteTable Yes
  RouteTableRef     FTPRtTbl
}

RouteTable FTPRtTbl
{
  Route Default 10.11.12.1 LMTULNK MTU 4096
}
```


This notes page shows an example of policy that could be written to solve the example problem, where the system programmer needed to move outbound FTP traffic being sent to a particular destination such that it uses a link to a network with a large MTU. This example assumes that all outbound FTP traffic sent to destination 10.11.33.1 is sent by jobs with names beginning with “FTP”.



For routing policies, you have the option of using a common configuration file that contains configuration used by multiple stacks, stack-specific configuration files that contain configuration used by only one stack, or a combination of the two.

If you want to use a common configuration file to configure Routing policy to be used by multiple stacks, you will configure the name of the file to policy agent using the new statement CommonRoutingConfig. This statement is specified in the main policy agent configuration file.

If you want to use a stack-specific configuration file to configure Routing policy to be used by only one stack, you will configure the name of the file to policy agent using the new statement RoutingConfig. This statement is specified in a policy agent image configuration file. If policy-based routing is to be used on a stack, the RoutingConfig statement MUST be specified for that stack. If all of the Routing policy configuration is contained in a common configuration file, specify the RoutingConfig statement without a path parameter.

When you want to use a common set of policies for multiple stacks then in the main policy agent configuration file specify the path of the common routing policy. For example ***CommonRoutingConfig /etc/routing_common.conf***. In each stack-specific configuration file specify **RoutingConfig** without a path name because only the common policy is needed. RoutingConfig, however, **must** be specified.

When you want to use totally unique policy for each stack then in the main configuration file do not specify a **CommonRoutingConfig** statement. In each stack-specific configuration file configure a RoutingConfig statement. For example ***RoutingConfig /etc/routing_stackxx.conf*** .

When you want to use a mix of common and stack-specific policies then in the main policy agent configuration file specify the path of the common routing policy. For example ***CommonRoutingConfig /etc/routing_common.conf***. In each stack-specific configuration file specify the path of the specific routing policy. For example ***RoutingConfig /etc/routing_stackxx.conf***.

When you want to have a mix of stacks with Routing policy and stacks without Routing policy then in the main policy agent configuration file specify the path of the common routing policy. For example ***CommonRoutingConfig /etc/routing_common.conf.*** In each stack-specific configuration file (only for stacks with Routing policy) specify the path of the routing policy for the stack. For example ***RoutingConfig /etc/routing_stackxx.conf.***

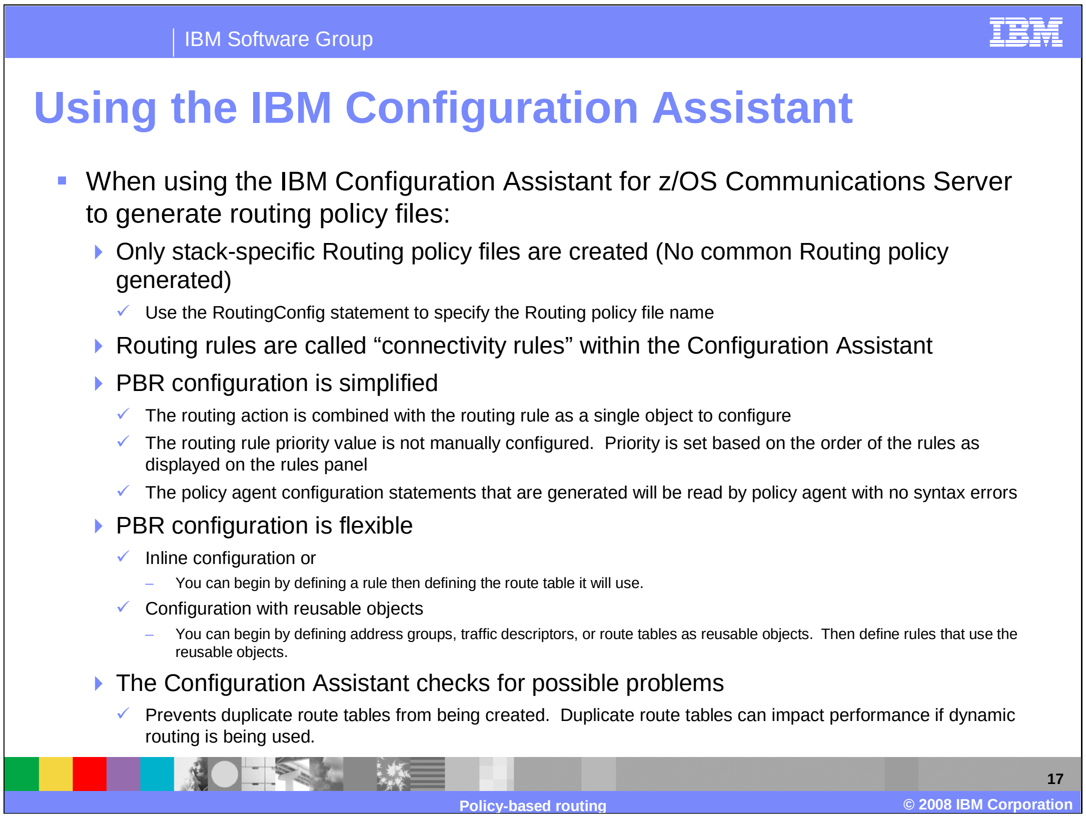
Routing policy deletion

- FLUSH
  - NOFLUSH option is not supported
  - Routing policies are always deleted before installing new policies at the following times:
    - Policy agent startup
    - TcpImage/PEPInstance statement added
    - MODIFY,REFRESH command issued
- NOPURGE
  - PURGE option is not supported
  - Routing policies never deleted during policy agent shutdown or when a TcpImage/PEPInstance statement is deleted
- To remove all routing policies from a TCP/IP stack:
  - Delete the RoutingConfig statement from the policy agent image configuration file for the stack

The policy agent FLUSH and NOFLUSH options indicate whether the policies of a particular policy type are deleted before installing new policies in the three scenarios listed on this page. Different combinations of FLUSH and NOFLUSH are supported across the policy types based upon the needs of each policy type. For Routing policies, only the FLUSH option is supported. All Routing policies are deleted prior to installing new Routing policies in these three scenarios.

The policy agent PURGE and NOPURGE options indicate whether the policies of a particular policy type are deleted in the two scenarios listed on this page. Different combinations of PURGE and NOPURGE are supported across the policy types based upon the needs of each policy type. For Routing policies, only the NOPURGE option is supported. Routing policies are not deleted in these two scenarios.

To remove all Routing policies that are installed in a stack you can delete the RoutingConfig statement from the image configuration file for the stack. If you are using centralized policy services, there are additional ways to remove all Routing policies that are installed in a stack. Refer to the presentation on centralized policy services for additional information.



As mentioned previously, the IBM Configuration Assistant for z/OS Communications Server can be used as an alternative to manually creating your policy agent configuration flat-file for policy-based routing. When this method is used, there are no common configuration files created. All configuration is placed in a stack-specific configuration file. Use the RoutingConfig statement in your image configuration file to specify the name and location of the file.

Within the Configuration Assistant, routing rules are called connectivity rules. This is the name that the Configuration Assistant uses for rules across the different types of policy.

Using the Configuration Assistant will simplify the job of configuring policy-based routing in a few ways.

First, there is no need to configure the policy rule and action as two separate objects. In the Configuration Assistant, these are configured as a single object that identifies both the characteristics of the traffic and the route tables that are to be used for that traffic.

Next, the rule priority does not need to be manually configured. Instead, the priority is managed by the Configuration Assistant based on the order that the rules are displayed on the rules panel.

Finally, the Configuration Assistant will verify the information you enter and will generate a flat-file containing policy statements that are free of syntax errors.

The Configuration Assistant will allow you flexibility in how you configure your Routing policy. You can configure everything inline for each rule and associated route table. Alternatively, you can create address groups, traffic descriptors, and route tables as reusable objects and then define rules that use these reusable objects.

If you use the Configuration Assistant to create your configuration files, it will ensure that you do not create duplicate route tables. Duplicate route tables should be avoided as they increase complexity and, in the case of route tables with dynamic routing support, they impact performance.

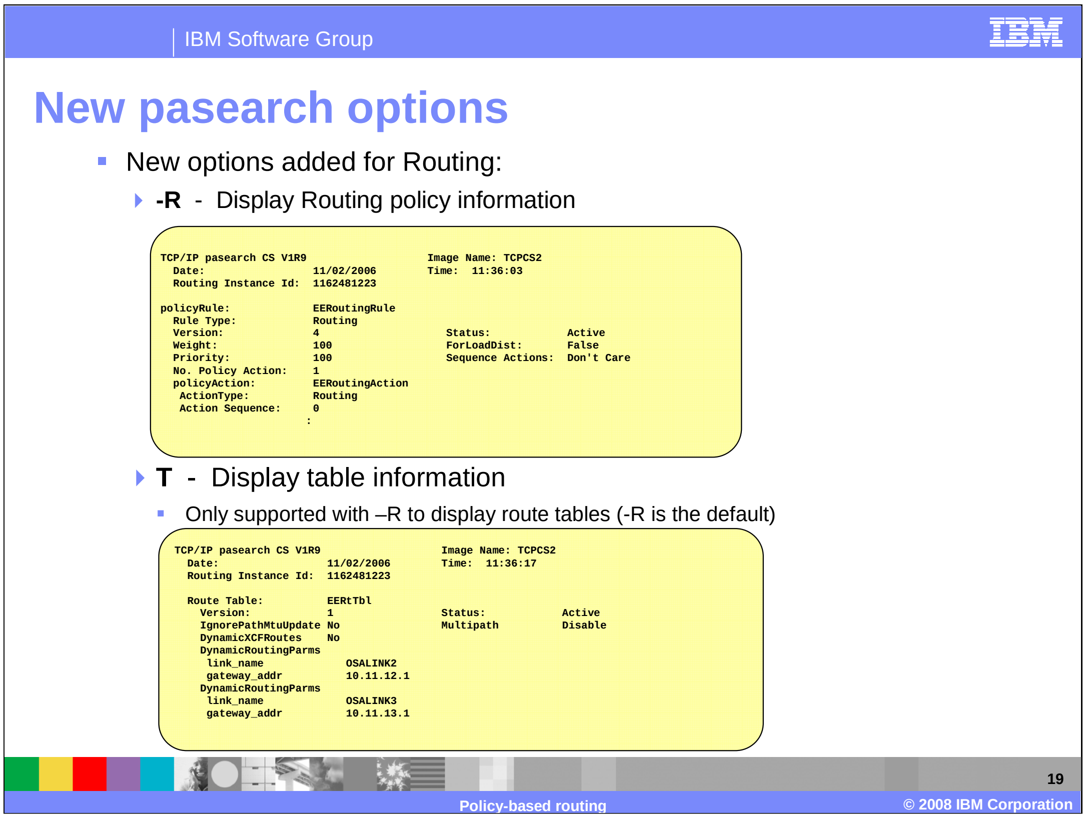
Stack and OMPROUTE config

- No configuration needed!
- All policy-based routing configuration is done in the policy agent flat-file
- The TCP/IP stack learns about the policy-based route tables, and the rules and actions for using them, from policy agent
- OMPROUTE learns about the policy-based route tables, and the parameters for controlling them, from the TCP/IP stack

Good news! There are no changes needed in either the TCP/IP stack configuration or the OMPROUTE configuration for policy-based routing.

The TCP/IP stack learns, from policy agent, about the policy rules, policy actions, and policy-based route tables that you have configured.

OMPROUTE learns, from the TCP/IP stack, about the policy-based route tables that you have configured to use dynamic routing.



In order to allow for querying of Routing policies, the set of options available for use with the pasearch command is expanded to include the -R and -T options. Use the -R option to indicate that the pasearch command is requesting Routing policy information. **Pasearch -R** will display active Routing rules and associated Routing actions. The active rule name displayed in this example is EERoutingRule. This active rule references the Routing action named EERoutingAction. The priority for this rule is 100. This active rule is installed in stack TCPCS2.

Use the -T option to indicate that the pasearch command is requesting table information. Currently, the -T option is only used to display policy-based route table information. This slide shows the display of an active Route table named EERtTbl. It is the result of a **pasearch -T -f EERtTbl** command**.** By using the pasearch -f option you can display a single Route table. This display indicates that two DynamicRoutingParms are defined for the table. Path MTU Update messages will be processed and the multipath routing algorithm is disabled for this routing table. In addition, direct routes to dynamic XCF addresses on other TCP/IP stacks will not be added to the route table.

These 2 new options can be used in combination with a variety of other pasearch options to control the information that will be displayed in response to the pasearch command. Following are some of the options that can be used in combination with the -R and -T options:

•Use -R with -e to display Routing policy rules and actions - this is the default.

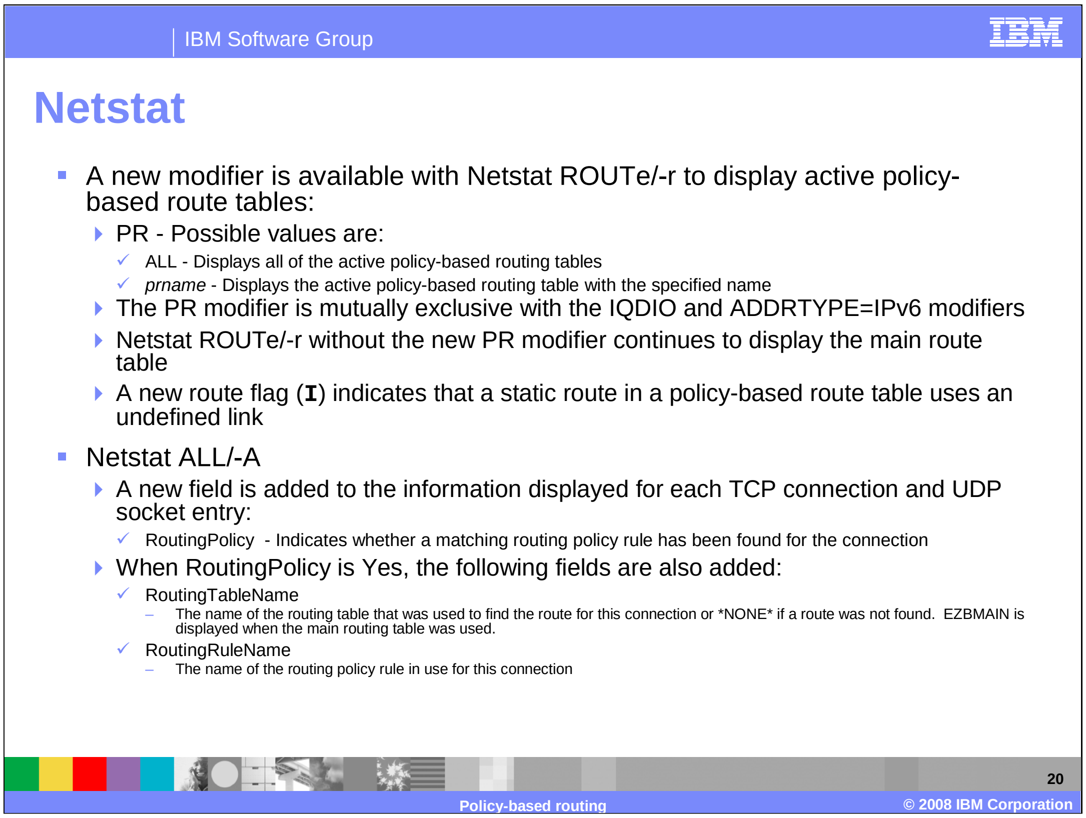
•Use -R with -r to display Routing policy rules.

•Use -R with -a to display Routing policy actions.

•Use -T with -R to display Routing route tables - this is the default.

Use -A with any of the combinations above to display active policy information - this is the default. Use **-I** with any of the combinations above to display inactive policy information. Use -f with any of the combinations above to filter the information displayed by policy name.

Note that a routing rule is active based on the IpTimeCondition. A routing action is always active. A route table is active if it is referenced by an active routing rule and its associated action.



A new modifier has been added to the Netstat ROUTe/-r command. The modifier is PR and it is used to indicate that active policy-based route tables are to be displayed. Since only active route tables are installed in the TCP/IP stack by policy agent, only active tables can be displayed by Netstat. The values that can be specified on the PR modifier are ALL or the name of a policy-based route table. Use ALL to request the display of all active policy-based route tables. Use the name of a policy-based route table to display only that active table.

The IQDIO modifier displays the HiperSockets Accelerator routing table, which is separate from the main route table and any policy-based route tables. Therefore, an error message will be issued if the PR modifier is used in combination with the IQDIO modifier.

Policy-based routing does not apply to IPv6 route tables. Therefore, no information will be displayed if the PR modifier is used in combination with the ADDRTYPE=IPV6 modifier.

A new flag has been added to the set of flags that can be displayed for each route included in the report. The I flag indicates a static route that is configured to use a link that is not defined to the stack.

The report generated by the Netstat ALL/-A command has been modified to include policy-based routing information for each TCP connection and UDP socket. The field RoutingPolicy indicates whether a matching routing policy rule has been found for the connection or socket entry. If so, the fields RoutingTableName and RoutingRuleName provide the names of the routing table and routing policy rule being used.

For an Enterprise Extender (EE) UDP socket entry, the RoutingPolicy value is always No. The routing policy information for an EE UDP socket entry is displayed using the DISPLAY NET,EEDIAG,TEST=YES command. For details on using this command, refer to z/OS Communications Server: SNA Operation.

NETSTAT ROUTE example

NETSTAT ROUTE PR prtable1

```
MVS TCP/IP NETSTAT CS V1R9       TCPIP Name: TCPCS           14:24:09
Policy Routing Table: prtable1
  IgnorePathMtuUpdate: Yes  MultiPath: Conn(Policy)
  DynamicXCFRoutes:    No
Destination      Gateway          Flags     Refcnt  Interface
-----------      -------          -----     ------  ---------
Default          9.67.115.65      UGS       000002  OSAQDIOLINK
9.67.115.65/32   0.0.0.0          UHS       000000  OSAQDIOLINK
9.67.115.69/32   0.0.0.0          UHS       000000  OSAQDIOLINK
9.67.113.0/24    0.0.0.0          SI        000000  OSALINK1
```


This example uses the PR modifier with the name of a policy-based route table. The resulting report includes the name of the route table, the three table-specific configuration values (IgnorePathMtuUpdate, MultiPath, and DynamicXCFRoutes), and the routes contained in the route table.

The MultiPath value shown in this example indicates that the perconnection multipath algorithm is to be used for the table. The value in the parentheses, Policy, indicates that this setting was configured on the policy RouteTable statement that defined this table. When the multipath setting for a policy-based route table is being inherited from the IPCONFIG MULTIPATH setting, the value in parentheses is Profile.

The last line of the report shows a static route that is defined using a link that is not currently defined to the stack. The I flag is used to indicate this.

When PR ALL is specified, similar information is repeated for all of the policy-based route tables.

Netstat ALL/-A example 1

- When no matching routing policy rule has been found for the connection

```
MVS TCP/IP NETSTAT CS V1R9       TCPIP NAME: TCPCS           17:40:36
Client Name: FTPD1                    Client Id: 0000003B
Local Socket: 0.0.0.0..21             Foreign Socket: 0.0.0.0..0
  Last Touched:       17:09:22          State:              Listen
  BytesIn:            0000000000        BytesOut:           0000000000
  SegmentsIn:         0000000000        SegmentsOut:        0000000000
  RcvNxt:             0000000000        SndNxt:             0000000000
  ClientRcvNxt:       0000000000        ClientSndNxt:       0000000000
      :                   :                 :                   :
      :                   :                 :                   :
  QOSPolicyRuleName:
  TTLSPolicy:         No
  RoutingPolicy:      No
  ReceiveBufferSize:  0000016384        SendBufferSize:     0000016384
  ConnectionsIn:      0000000000        ConnectionsDropped: 0000000000
  CurrentBacklog:     0000000000        MaximumBacklog:     0000000010
  CurrentConnections: 0000000300        SEF:                098
      :                   :                 :                   :
      :                   :                 :                   :
```


This example shows a connection for which a matching routing policy rule has not been found. Note that the RoutingTableName and RoutingRuleName fields are not displayed.

Netstat ALL/-A example 2

- When a matching routing policy rule has been found for the connection

```
MVS TCP/IP NETSTAT CS V1R9       TCPIP NAME: TCPCS           17:40:36
Client Name: FTPD1                    Client Id: 0000003B
Local Socket: 0.0.0.0..21             Foreign Socket: 0.0.0.0..0
  Last Touched:       17:09:22          State:             Listen
  BytesIn:            0000000000        BytesOut:          0000000000
  SegmentsIn:         0000000000        SegmentsOut:       0000000000
  RcvNxt:             0000000000        SndNxt:            0000000000
  ClientRcvNxt:       0000000000        ClientSndNxt:      0000000000
      :                   :                  :                  :
      :                   :                  :                  :
  QOSPolicyRuleName:
  TTLSPolicy:         No
  RoutingPolicy:      Yes
    RoutingTableName: PRTAB1
    RoutingRuleName:  SecLow2
  ReceiveBufferSize:  0000016384        SendBufferSize:     0000016384
  ConnectionsIn:      0000000000        ConnectionsDropped: 0000000000
  CurrentBacklog:     0000000000        MaximumBacklog:     0000000010
  CurrentConnections: 0000000300        SEF:                098
      :                   :                  :                  :
      :                   :                  :                  :
```


This example shows a connection for which a matching routing policy rule has been found. The RoutingTableName and RoutingRuleName fields are now displayed, showing the route table and routing policy rule being used.

Netstat ALL/-A example 3

- When the route tables referenced by the matching routing policy rule do not contain a usable route to the destination

```
MVS TCP/IP NETSTAT CS V1R9       TCPIP Name: TCPCS1          20:46:31
Client Name: USER105                  Client Id: 0000004D
  Local Socket: 10.11.2.1..1024
  Foreign Socket: 10.81.2.2..4006
    BytesIn:            00000000000000000005
    BytesOut:           00000000000000000010
    SegmentsIn:         00000000000000000003
    SegmentsOut:        00000000000000000005
    Last Touched:       20:45:04          State:              Establsh
    RcvNxt:             2928345537        SndNxt:             2928339715
    ClientRcvNxt:       2928345537        ClientSndNxt:       2928339715

    QOSPolicyRuleName:
    RoutingPolicy:      Yes
      RoutingTableName: *NONE*
      RoutingRuleName:  RoutingRule1
    ReceiveBufferSize:  0000016384        SendBufferSize:     0000016384
    ReceiveDataQueued:  0000000000
    SendDataQueued:     0000000000
```


This example shows a connection for which a matching routing policy rule has been found. However, a search of all of the associated route tables has failed to find a usable route to the destination. When this is the case, the RoutingTableName is displayed as *NONE*.

OMPROUTE DISPLAY options

- New options available for displaying policy-based route tables with the OMPROUTE display command (DISPLAY TCPIP,*tcpipjobname*,OMProute,RTTABLE):
  - PRtable=ALL
    - Displays the routes in all of the OMPROUTE policy-based route tables, along with the dynamic routing parameters for each table
  - PRtable=*prname*
    - Displays the routes in the specified OMPROUTE policy-based route table, along with the dynamic routing parameters for the table
- The DEST=*ip_addr* option can be used with both PRtable=ALL and PRtable=*prname* to display the routes to a single destination
- Only route tables with dynamic routing parameters defined can be displayed.

A new option has been added to the OMPROUTE DISPLAY RTTABLE command. The option is PRtable and it is used to indicate that OMPROUTE policy-based route tables are to be displayed. The values that can be specified on the PRtable option are ALL or the name of a policy-based route table. Use ALL to request the display of all OMPROUTE policy-based route tables. Use the name of a policy-based route table to display only that table.

The PRtable option can be used in combination with the DEST= option to display details of the routes in policy-based route tables to a particular destination.

OMPROUTE has no knowledge of policy-based route tables that are defined without dynamic routing parameters. Those route tables are using static routing only. Since OMPROUTE has no knowledge of those tables, they cannot be displayed with the OMPROUTE DISPLAY command.

OMPROUTE display - Example1

DISPLAY TCPIP,*tcpipjobname*,OMProute,RTTABLE,PRtable=SECLOW2

```
EZZ7847I ROUTING TABLE 796
TABLE NAME:     SECLOW2
TYPE   DEST NET         MASK      COST    AGE     NEXT HOP(S)

SBNT   8.0.0.0          FF000000  1       1549    NONE
 SPF   8.8.8.8          FFFFFFFC  2       1545    9.67.100.8
 SPF   8.8.8.8          FFFFFFFF  2       1545    9.67.100.8
SBNT   9.0.0.0          FF000000  1       1368    NONE
 DIR*  9.67.100.0       FFFFFF00  1       1576    9.67.100.7
 SPF   9.67.100.7       FFFFFFFF  2       1545    OSALINK2
 SPF   9.67.100.8       FFFFFFFF  1       1572    9.67.100.8
 SPF   9.67.105.4       FFFFFFFF  2       1545    9.67.100.8
SPE2   130.200.0.0      FFFF0000  0       1379    9.67.100.8 (2)
                       0 NETS DELETED

DYNAMIC ROUTING PARAMETERS:
  INTERFACE:   OSALINK2      NEXT HOP: 9.67.100.8
  INTERFACE:   OSALINK2      NEXT HOP: 9.67.100.15
  INTERFACE:  *OSALINK3      NEXT HOP: 9.67.201.53
```


In this example, the PRtable=*prname* option has been used to display a particular policy-based route table. The *prname* value of SECLOW2 results in only that route table being displayed.

Table SECLOW2 is defined with three dynamic routing parameters that each specify a link and next hop. All dynamic routes added to this table should be either direct routes over one of these links or indirect routes over one of the links that have the associated IP address as next hop.

Most of the information in the display of a policy-based route table is the same as what is included in the display of the main route table. What is added for policy-based route tables is the name of the table at the top and the dynamic routing parameters being used for the table at the bottom.

The asterisk beside the link name in the last dynamic routing parameter shown in this example indicates that OSALINK3 is either not currently defined to the TCP/IP stack or not currently active. In either case, there would be no dynamic routes in the route table over that link.

When PRtable=ALL is specified, similar information is repeated for all of the OMPROUTE policy-based route tables.

OMPROUTE display example2

DISPLAY TCPIP,*tcpipjobname*,OMProute,RTTABLE,PRtable=SECLOW2,DEST=130.200.0.0

```
EZZ7874I ROUTE EXPANSION 370
TABLE NAME:      SECLOW2
DESTINATION:     130.200.0.0
MASK:            255.255.0.0
ROUTE TYPE:      SPE2
DISTANCE:        0
AGE:             1385
NEXT HOP(S):     9.67.100.8        (OSALINK2)
                 9.67.100.15       (OSALINK2)
```


In this example, the DEST=*ip_addr* option has been used in order to display the multiple next hops to the 130.200.0.0 network that appear in policy-based route table SECLOW2.

If PRtable=ALL was specified instead of PRtable=SECLOW2 and other policy-based route tables also contained routes to 130.200.0.0, the information for the routes in each route table would be included in the report.

Display NET,EEDIAG command

- z/OS VTAM® display NET,EEDIAG,TEST=YES command is modified as follows:
  - The policy-based routing information associated with the EE connectivity being tested is displayed
    - ✓ Route table and Routing Rule
  - The number of IP routes tested between two EE endpoints now defaults to 16 - can be altered with the new **MAXROUTE** operand
  - Message IST2139I message is modified to display total number of routes tested and the overall number of valid routes found

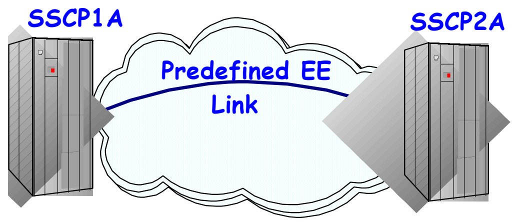
The Display EEDIAG,TEST=YES command, or Enterprise Extender (EE) connectivity test command, is useful in debugging various network problems and was introduced in z/OS V1R8 Communications Server. This command can be used to test an existing Enterprise Extender connection, or it can be used to assist in diagnosing why an EE connection cannot be established.

With policy-based routing, each of the EE ports can be associated with a unique routing rule. The EE traffic utilizing each port could be routed using different route tables. Due to these changes, the command required modifications for the support of policy-based routing.

The MAXROUTE value specifies the maximum number of valid TCP/IP routes that will be tested between two Enterprise Extender (EE) endpoints. Multiple routes may exist when MULTIPATH support or policy-based routing is being used in the route calculations between the EE endpoints. When the maximum routes to be tested is exceeded then all routes over the limit will not be tested.

Normally 16 routes (the default) should be sufficient to fully test connectivity between two EE endpoints. If message IST2139I indicates that all routes are not being tested, MAXROUTE can be used to increase the number of routes to be tested.

Successful single hop EE connectivity test




```
XCAIP    VBUILD TYPE=XCA
PORTIP   PORT   MEDIUM=HPRIP,
                LIVTIME=(25,3600),
                SRQTIME=15,
                SRQRETRY=3
*
GPIP     GROUP  DIAL=YES,
                ANSWER=ON
                    ISTATUS=INACTIVE,
                IPADDR=9.67.1.1,
                CALL=INOUT
LNIP1    LINE
PUIP1    PU
LNIP2    LINE
PUIP2    PU
```

```
TOIP2A   VBUILD TYPE=SWNET
*
SWPU1A2A PU     ADDR=01,
                CPCP=YES,
                CPNAME=SSCP2A,
                CONNTYPE=APPN,
                NETID=NETA,
                PUTYPE=2
PATH1A2A PATH   GRPNM=GPIP,
                IPADDR=9.67.1.2
```


This is a sample of a single hop predefined Enterprise Extender (EE) link. This example illustrates a case where connectivity being tested between two EE endpoints is successful.

IBM Software Group

IBM

Successful single hop EE connectivity test

```
D NET,EEDIAG,TEST,ID=SWPU1A2A,DETAIL
 IST097I DISPLAY ACCEPTED
 IST350I DISPLAY TYPE = EEDIAG
 IST2119I ENTERPRISE EXTENDER DISPLAY CORRELATOR: EE000001
 IST2067I EEDIAG DISPLAY ISSUED ON 03/13/05 AT 21:07:01
 IST1680I LOCAL IP ADDRESS 9.67.1.1
 IST1680I REMOTE IP ADDRESS 9.67.1.2
 IST2023I CONNECTED TO LINE LNIP2
 IST2126I CONNECTIVITY TEST IN PROGRESS
 IST314I END
 IST350I DISPLAY TYPE = EEDIAG
 IST2130I ENTERPRISE EXTENDER CONNECTIVITY TEST INFORMATION
 IST2119I ENTERPRISE EXTENDER DISPLAY CORRELATOR: EE000001
 IST2131I EEDIAG DISPLAY COMPLETED ON 03/13/05 AT 21:07:46
 IST2132I LDLC PROBE VERSIONS: VTAM = V1  PARTNER = UNKNOWN
 IST1680I LOCAL IP ADDRESS 9.67.1.1
 IST1680I REMOTE IP ADDRESS 9.67.1.2
 IST2224I ENTERPRISE EXTENDER ROUTING POLICY INFORMATION
 IST2225I  PORT    ROUTE TABLE   ROUTING RULE
 IST2205I  ----    -----------   ------------
 IST2226I  12000   EERTTBL       EEROUTINGRULE
 IST2226I  12001   EERTTBL       EEROUTINGRULE
 IST2226I  12002   EERTTBL       EEROUTINGRULE
 IST2226I  12003   EERTTBL       EEROUTINGRULE
 IST2226I  12004   EERTTBL       EEROUTINGRULE
 IST924I -------------------------------------------------------------
 IST2134I   CONNECTIVITY SUCCESSFUL                        PORT: 12000
 IST2137I     1  9.67.1.2                   RTT:     6
 IST2134I   CONNECTIVITY SUCCESSFUL                        PORT: 12001
 IST2137I     1  9.67.1.2                   RTT:     6
 IST2134I   CONNECTIVITY SUCCESSFUL                        PORT: 12002
 IST2137I     1  9.67.1.2                   RTT:     6
 IST2134I   CONNECTIVITY SUCCESSFUL                        PORT: 12003
 IST2137I     1  9.67.1.2                   RTT:     6
 IST2134I   CONNECTIVITY SUCCESSFUL                        PORT: 12004
 IST2137I     1  9.67.1.2                   RTT:     7
 IST924I -------------------------------------------------------------
 IST2139I CONNECTIVITY TEST INFORMATION DISPLAYED FOR 1 OF 1 ROUTES
```

- All EE traffic uses single policy-based routing rule and route table

30

Policy-based routing

In this example, the operator on SSCP1A performs the EE connectivity test to assist in determining the connectivity to the remote EE endpoint located on SSCP2A.

A single policy-based routing rule (EEROUTINGRULE) is being used for all EE traffic between the EE endpoints being tested. The policy-based routing rule has indicated that Multipath is disabled and that there is a RouteTable defined for EE traffic (the route table name is EERTTBL).

When a policy-based routing rule is defined for any EE traffic between the EE endpoints, you will receive additional messages. Message IST2224I is a header message for the routing policy information.

Message IST2225I is a header for the display of the EE UDP ports, route tables, and policy routing rules. Message IST2226I displays the EE UDP ports and each port’s associated route table and policy routing rule. If a policy-based routing rule is not defined for an EE UDP port, the policy routing rule will be displayed as NONE. When the main route table is being utilized (either no policy routing rule or the routing action indicates the use of the main routing table), the route table that is displayed is EZBMAIN.

Message IST2139I has been modified to indicate the number of routes tested and the overall number of routes found for the EE connectivity test. If this message indicates that all routes are not being tested then the D NET,EEDIAG,TEST command can be re-issued with a MAXROUTE operand value sufficiently large enough to test all routes.

Diagnosis

- Possible configuration problems
  - Policy agent log files
  - Console log
  - Use pasearch to view the routing rules, routing actions and route tables configured in policy
- Connectivity problem between policy agent and the TCP/IP stack
  - Provide TCP/IP CTRACE (POLICY, INTERNET, & IOCTL options)
- Problems with IP Routing
  - Use updated Netstat commands to display policy-based routing information
  - Use TCP/IP CTRACE with options POLICY and INTERNET
  - When processing a dump with TCPIPCS:
    - ✓ TCPIPCS ROUTE
      - PR option displays policy-based route tables
      - PD option displays policy-based route tables that have been recently deleted from policy
    - ✓ TCPIPCS POLICY
      - Updated to include routing policy information installed in the stack
- Problems with Dynamic Routing
  - Use updated OMPROUTE Display commands to view the policy-based route tables that are being managed by OMPROUTE
  - Many OMPROUTE messages have been updated to include the name of the route table being processed
- A new SNMP MIB object is added to the IBM TCP/IP enterprise-specific MIB module
  - Indicates whether a route is from a policy-based route table

If you experience a problem that you believe may be caused by the processing of the policy configuration files by policy agent or the installation of the policies into the TCP/IP stack, you can use the pasearch command to view the policies as they are known by policy agent. The console log and the policy agent log files are useful in verifying each configuration statement that was processed, any errors encountered, and each policy that was installed in the TCP/IP stack. A policy agent log file can be specified using the -l (lower case L) start option. If not specified, the default log file is /tmp/pagent.log. Policy agent should be run with a Log Level of at least 127. If there is a policy problem, start policy agent with -d 32 which will log policies sent from policy agent to the stack.

For problem determination, the documentation provided should include any dumps generated, the policy agent configuration files if the problem is configuration related, and the TCP/IP CTRACE if the problem seems to be in the communication between policy agent and the TCP/IP stack.

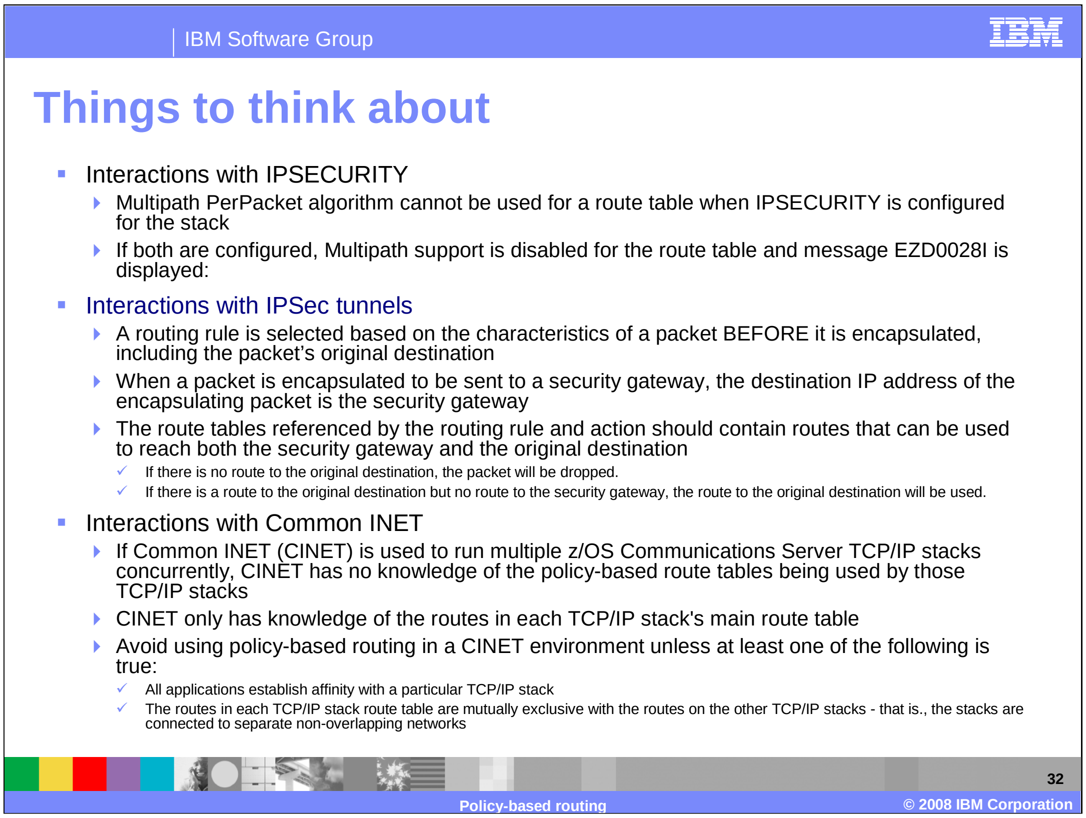
If you experience a problem that you believe may be caused by the contents of the policy-based route tables in the stack or the way in which IP Routing is using the route tables, you can use the Netstat command to view the route tables and the policy rule and route table being used by each connection. For problem determination, the documentation provided should include the TCP/IP CTRACE with options POLICY and INTERNET active.

If you have a dump of the TCP/IP address space, you can use enhancements in TCPIPCS ROUTE and TCPIPCS POLICY to access policy-based routing data. The PR and PD options of TCPIPCS ROUTE allow for the display of policy-based route tables. The TCPIPCS POLICY report includes any Routing policy information installed in the stack.

If you experience a problem that you believe may be caused by the management of the dynamic routes in policy-based route tables by OMPROUTE, you can use the OMPROUTE Display command to view the versions of the route tables within OMPROUTE.

When diagnosing an OMPROUTE problem by reviewing OMPROUTE messages on the MVS console or in the OMPROUTE debug files, you will notice that many of the OMPROUTE messages are updated to include the name of the associated route table. This change will be seen even if the policy-based routing function is not being used. In this case, all of the changed messages will now include the name of the main route table, EZBMAIN.

The IBM TCP/IP enterprise-specific MIB module is updated to support policy-based routing. The MIB object ibmMvsRouteFlags has been added to indicate whether a route is from a policy-based route table.



As with the main route table, the Multipath PerPacket algorithm cannot be used with a policy-based route table when IPSECURITY is configured for the stack. If these two are configured together, the Multipath function will be disabled.

If IPSECURITY is in use on a stack, use either no Multipath algorithm or the Multipath PerConnection algorithm to distribute traffic routed using your policy-based route tables. If Multipath PerPacket is configured in the policy-based route table when IPSECURITY is configured for the stack then this message is displayed:

EZD0028I IPV4 MULTIPATH PERPACKET NOT VALID WITH IPV4 SECURITY – MULTIPATH SUPPORT DISABLED FOR ROUTE TABLE *table*

When policy-based routing is used on a stack that is also using IP Security, special care must be taken to ensure that policy-based route tables that will be used for IP Security traffic contain the necessary routes. The route tables to be searched will be selected based on the characteristics of the unencapsulated packet. The destination of the encapsulated packet may differ from the destination of the original packet. Routes to both destinations should appear in the route tables to be searched.

The use of policy-based routing on a stack that is in a Common INET environment should be avoided except in very limited scenarios. Those scenarios are described on this page. The problem with this combination in other scenarios is due to the fact that Common INET has no knowledge of the policy-based route tables being used by each stack. Common INET only knows the contents of each stack's main route table and will select a stack to used for a connection based on that information. Since the contents of policy-based route tables will likely differ from the contents of the main route table, Common INET may not select the best stack for the connection and may even select a stack on which the connection will fail.

Performance considerations

- Dynamic routes are maintained for each policy-based route table that is using dynamic routing (that is, DynamicRoutingParms are specified in the route table configuration)
- There is an OMPROUTE performance cost for each table using dynamic routing
- Avoid large numbers of policy-based route tables using dynamic routing
- Avoid duplicate route tables

Each policy-based route table that is configured for dynamic routing adds additional processing to OMPROUTE. Duplicate route tables should be avoided and this is ensured if the Configuration Assistant is used to create your policy agent flat-files. In addition, you should avoid having large numbers of policy-based route tables that use dynamic routing.

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_Pol_Based_Routing.ppt

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

AT IBM VTAM z/OS

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

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

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

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

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

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

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.