Router Teldat

Policy-Based Routing

Doc. *DM745-I* Rev. *10.10*
April 2003

INDEX

Chapter 1
Policy-Based Routing Technology

1. Introduction

In today's high performance internetworks, organizations need the freedom to implement packet forwarding and routing according to their own defined policies in a way that goes beyond traditional routing protocol concerns. Where administrative issues dictate that traffic be routed through specific paths, policy-based routing, introduced in Teldat Routing Software Release 10.1, can provide the solution. By using policy-based routing, customers can implement policies that selectively cause packets to take different paths.

Policy routing also provides a mechanism to mark packets so that certain kinds of traffic receive differentiated, preferential service when used in combination with queuing techniques. These queuing techniques provide an extremely powerful, simple, and flexible tool to network managers who implement routing policies in their networks.

This chapter discusses the Teldat routing software policy-based routing feature and addresses policy-based routing and its functionality. In addition, the issues related to managing an internetwork with policy-based routing implemented are described. And finally, the applications of policy-based routing in internetworks are presented.

2. The Benefits of Policy-Based Routing

The benefits that can be achieved by implementing policy-based routing in the networks include:

- Source-Based Transit Provider Selection — Internet service providers and other organizations can use policy-based routing to route traffic originating from different sets of users through different Internet connections across the policy routers.
- Quality of Service (QOS) — Organizations can provide QOS to differentiated traffic by setting the precedence or type of service (TOS) values in the IP packet headers at the periphery of the network and leveraging queuing mechanisms to prioritize traffic in the core or backbone of the network.
- Cost Savings — Organizations can achieve cost savings by distributing interactive and batch traffic among low-bandwidth, low-cost permanent paths and high-bandwidth, high-cost, switched paths.
- Load Sharing — In addition to the dynamic load-sharing capabilities offered by destination-based routing that the Teldat routing software has always supported, network managers can now implement policies to distribute traffic among multiple paths based on the traffic characteristics.

3. Policy-Based Routing Data Forwarding

Policy-based routing (PBR) provides a mechanism for expressing and implementing forwarding/routing of data packets based on the policies defined by the network administrators. It provides a more flexible mechanism for routing packets through routers, complementing the existing mechanism provided by routing protocols.

Routers forward packets to the destination addresses based on information from static routes or dynamic routing protocols such as Routing Information Protocol (RIP) or Open Shortest Path First (OSPF). Instead of routing by the destination address, policy-based routing allows network administrators to determine and implement routing policies to allow or deny paths based on the following:

- Identity of a particular end system
- Application
- Protocol
- Size of packets

Policies can be defined as simply as "my network will not carry traffic from the engineering department" or as complex as "traffic originating within my network with the following characteristics will take path A, while other traffic will take path B."

3.1. Tagging Network Traffic

Policy-based routing allows the network administrator to classify traffic using access control lists (ACLs) and then set the IP precedence or TOS values, thereby tagging the packets with the defined classification.

Classification of traffic through policy-based routing allows the network administrator to identify traffic for different classes of service at the perimeter of the network and then implement QOS defined for each class of service in the core of the network using priority, custom, or weighted fair queuing techniques. This process saves having to classify the traffic explicitly at each WAN interface in the core/backbone network.

4. Applying Policy-Based Routing

Policy-based routing is applied to incoming packets. All packets received on an interface with policy-based routing enabled are considered for policy-based routing. The router passes the packets through enhanced packet filters called route maps. Based on the criteria defined in the route maps, packets are forwarded/routed to the appropriate next hop.

4.1. Policy Route Maps

Each entry in a route map statement contains a combination of match and set clauses/commands. The match clauses define the criteria for whether appropriate packets meet the particular policy (that is, the conditions to be met). The set clauses explain how the packets should be routed once they have met the match criteria.

For each combination of match and set commands in a route map statement, all sequential match clauses must be met simultaneously by the packet for the set clauses to be applied. There may be multiple sets of combinations of match and set commands in a full route map statement.

The route map statements can also be marked as permit or deny. If the statement is marked as a deny, the packets meeting the match criteria are sent back through the normal forwarding channels (in other words, destination-based routing is performed). Only if the statement is marked as permit and the packets meet the match criteria are all the set clauses applied. If the statement is marked as permit and the packets do not meet the match criteria, then those packets are also forwarded through the normal routing channel.

Policy routing is specified on the interface that receives the packets, not on the interface from which the packets are sent.

4.2. Match Clauses — Defining the Criteria

The IP standard or extended ACLs can be used to establish the match criteria. The standard IP access lists can be used to specify the match criteria for source address; extended access lists can be used to specify the match criteria based on application, protocol type, TOS, and precedence.

The match clause feature has been extended to include matching packet length between specified minimum and maximum values. The network administrator can then use the match length as the criterion that distinguishes between interactive and bulk traffic (bulk traffic usually has larger packet sizes).

The policy routing process proceeds through the route map until a match is found. If no match is found in the route map, or the route map entry is made a deny instead of a permit, then normal destination-based routing of the traffic ensues.

There is an implicit deny at the end of the list of match statements.

4.3. Set Clauses — Defining the Route

If the match clauses are satisfied, one of the following set clauses can be used to specify the criteria for forwarding packets through the router; they are evaluated in the order listed:

1. List of specified IP addresses and/or interfaces through which the packets can be routed — The IP address can specify the adjacent next hop router in the path toward the destination to which the packets should be forwarded. The first IP address associated with a currently "up" connected interface, or the first specified interface that is found to be up, whichever comes first, will be used to route the packets.
2. List of default IP addresses and/or interfaces — Route to the interface or the next hop specified by this set clause only if there is no explicit route for the destination address of the packet in the routing table. The first IP address associated with a currently "up" connected interface, or the first specified interface that is found to be up, whichever comes first, will be used to route the packets.
3. IP TOS — A value or keyword can be specified to set the type of service in the IP packets.
4. IP precedence — A value or keyword can be specified to set the precedence in the IP packets.

The set commands can be used in conjunction with each other.

If the packets do not meet any of the defined match criteria (that is, if the packets fall off the end of a route map), then those packets are routed through the normal destination-based routing process. If it is desired not to revert to normal forwarding and to drop the packets that do not match the specified criteria, then a Loopback interface should be specified as the last interface in the list by using the set clause.

4.4. Management Implications

The route specified by configured policies might differ from the best route as determined by the routing protocols, enabling packets to take different routes depending on their source, length, and content. As a result, packet forwarding based on configured policies will override packet forwarding based on the routing entries in the routing tables to the same destination. For example, the management applications might discover a path that will pertain to the path discovered by a dynamic routing protocol or specified by static route mapping, whereas the actual traffic might not follow that path, based on the configured policies.

Similarly, the traceroute command might generate a path that is a different from the route taken by the packets generated by the user application.

Because the added flexibility to route traffic on user-defined paths rather than the paths determined by routing protocols may make the environment more difficult to manage and might cause routing loops, policies should be defined in a deterministic manner to keep the environment simple and manageable.

Chapter 2
Policy Routing Configuration

1. Introduction

To enable policy routing, you must identify which route map to use for policy routing and create the route map. The route map itself specifies the match criteria and the resulting action if all of the match clauses are met. These steps are described in the following task tables.

To define the route map to be used for policy routing, use the following command in global configuration mode:

Command	Aim
`Config>`**`feature route-map`**	Enters route map configuration mode.
`Route map config>`**`route-map`** *`map-tag`*	Defines a route map to control where packets are output.

Then configure the match and set clauses to define the criteria by which packets are examined to learn if they will be policy-routed, and to set the actions to be taken on matching packets.

To enable policy routing on an interface, indicate which route map the router should use by using the following command in IP protocol configuration menu. All packets arriving on the specified interface will be subject to policy routing.

Command	Aim
`IP config>`**`policy`** *`interface`* **`route-map`** *`map-tag`*	Identifies the route map to use for policy routing packets arriving on an interface.
`IP config>`**`local policy route-map`** *`map-tag`*	Identifies the route map to use for policy routing locally generated packets.

2. Defining the route map

A route map is a profile indicating the criteria through which traffic is classified and the actions to be executed over the traffic selected by the said criteria. Each interface can be associated to a route map to execute policy routing and various interfaces can share the same route map. You can also specify a route map to carry out policy routing over the packets generated by the device itself.

In order to configure a route map, first access the general configuration menu:

```
*PROCESS 4
Config>
```

From this general configuration menu access the route maps configuration menu:

```
Config>FEATURE ROUTE-MAP

-- Route maps user configuration --
Route map config>
```

The following commands are available from the route maps configuration menu:

Command	Aim
list	Displays the names of the defined route maps.
no route-map *map-id*	Deletes a route map.
route-map *map-id*	Defines a route map.
exit	Returns to the general configuration menu.

The following sections provide a more in-depth explanation for these commands.

2.1. LIST

This command displays a list of all the defined route map names.

Syntax:

```
Route map config>LIST
```

Example:

```
Route map config>LIST

Configured route maps:
  route-map admin
  route-map office
Route map config>
```

2.2. NO ROUTE-MAP

This command completely eliminates the definition of a route map.

Syntax:

```
Route map config>NO ROUTE-MAP <map-id>
```

map-id	Name of the route map to be eliminated.

Example:

```
Route map config>NO ROUTE-MAP office
Route map config>
```

2.3. ROUTE-MAP

This command enters the specified route map configuration mode.

Syntax:

```
Route map config>ROUTE-MAP <map-id>
```

map-id	Route map name to define.

Example:

```
Route map config>ROUTE-MAP OFFICE


Route map office>
```

A route map is made up of one or more numerically ordered entries.

When a packet is received through an interface, the device accesses the route map to which the said interface is associated to and checks the packet successively with each entry until it finds one that matches (i.e. the complies with the match clauses).

If no entry matching the packet is found, then this is conventionally routed complying with the routes table.

If an entry matching the packet is found and this is in deny mode, then the packet is conventionally routed complying with the routes table.

If an entry matching the packet is found and this is in permit mode (default mode), then the set clauses for the said entry are applied.

The route map configuration menu has the following commands:

Command	Aim
entry *n* **default**	Establishes the default configuration in the specified entry.
entry *n* **deny**	Packets matching this entry are routed through the routes table (without policy routing).
entry *n* **match ip address** {*access-list*} [*...access-list*]	Determines the access lists to be used in order to know if a packet matches the entry.
entry *n* **match length** *minimum-length maximum-length*	Establishes a minimum and maximum packet length in order to match the entry.

entry *n* **permit**	Packets matching this entry have the set clauses applied to them. This is the default mode.
entry *n* **set ip default next-hop** {*ip-address* | *interface*} [...*ip-address* | ...*interface*]	Defines a list of ip addresses and/or interfaces to route the packet if no specified route is found.
entry *n* **set ip next-hop** {*ip-address* | *interface*} [...*ip-address* | ...*interface*]	Defines a list of ip addresses and/or interfaces in order to route the packet.
entry *n* **set ip precedence** *precedence*	Establishes the ip header precedence field.
entry *n* **set ip tos** *service-type*	Establishes the ip header tos field.
no entry *n*	Deletes an entry.
no entry *n* **match ip address**	Deletes all the access lists from the check list.
no entry *n* **match ip address** {*access-list*} [...*access-list*]	Deletes the specified access lists from the check list.
no entry *n* **match length**	Eliminates the packet length check.
no entry *n* **set ip default next-hop**	Eliminates the default next hop list.
no entry *n* **set ip default next-hop** {*ip-address* | *interface*} [...*ip-address* | ...*interface*]	Eliminates the specified ip addresses /interface from the default next hop list.
no entry *n* **set ip next-hop**	Eliminates the next hop list.
no entry *n* **set ip next-hop** {*ip-address* | *interface*} [...*ip-address* | ...*interface*]	Eliminates the specified ip addresses /interface from the next hop list.
no entry *n* **set ip precedence**	Does not modify the ip header precedence field.
no entry *n* **set ip tos**	Does not modify the ip header tos field.

The following paragraphs explain these commands:

a) ENTRY n DEFAULT

This command establishes the default configuration in the specified entry.

Syntax:

```
Route map config>ENTRY n DEFAULT
```

n	Entry number. The entries do not have to be consecutively numbered, however the number order is important as the lowest number is checked first. The range is from 1 to 65535.

Example:

```
Route map office> ENTRY 1 DEFAULT
Route map office>
```

b) ENTRY n DENY

This command establishes the deny mode in an entry. The packets matching this entry are conventionally routed through the routes table and the set clause is not applied.

Syntax:

```
Route map config>ENTRY n DENY
```

n	Entry number. The entries do not have to be consecutively numbered, however the number order is important as the lowest number is

	checked first. The range is from 1 to 65535.

Example:

```
Route map office>ENTRY 1 DENY
Route map office>
```

c) ENTRY n PERMIT

This command establishes the permit mode in an entry. Packets matching this entry will have the set clause applied. This is the default mode for an entry.

Syntax:

```
Route map config>ENTRY n PERMIT
```

n	Entry number. The entries do not have to be consecutively numbered, however the number order is important as the lowest number is checked first. The range is from 1 to 65535.

Example:

```
Route map office>ENTRY 1 PERMIT
Route map office>
```

d) ENTRY n MATCH IP ADDRESS

This command determines the access lists to be used in order to discern if a packet matches the entry.

The access lists associated through this command must have been previously created and cannot be associated with any other protocol.

The access lists are checked in the order in which they are associated with this command.

If the packet does not match one access list, the next list is checked.

If the packet matches an access list and this is permit, then the packet matches the route map entry.

If the packet matches an access list and this is deny, then the packet does not match the route map entry and this passes to check the subsequent route map entry.

Syntax:

```
Route map config>ENTRY n MATCH IP ADDRESS PERMIT{access-list} [...access-list]
```

n	Entry number. The entries do not have to be consecutively numbered, however the number order is important as the lowest number is checked first. The range is from 1 to 65535.
access-list	Access control list number. You can specify both standard as well as extended lists. The range is from 1 to 199.

Example:

```
Route map office>ENTRY 1 MATCH IP ADDRESS 101
Route map office>
```

e) ENTRY n MATCH LENGTH

This command sets the maximum and minimum packet lengths in order to match the entry.

This length considered is level 3 i.e. the total length of the IP packet (header plus data). This length is given in the IP header field.

Syntax:

```
Route map config>ENTRY n MATCH LENGTH minimum-length maximum-length
```

n	Entry number. The entries do not have to be consecutively numbered, however the numerical order is important as the lowest number is checked first. The range is from 1 to 65535.
minimum-length	If the packet length is less than this then it will not match the entry. The range is from 0 to 4294967295.
maximum-length	If the packet length is greater than this then it will not match the entry. The range is from 0 to 4294967295.

Example:

```
Route map office>ENTRY 1 MATCH LENGTH 0 127
Route map office>
```

f) ENTRY n SET IP DEFAULT NEXT-HOP

This command determines the ip address or interface to use in order to route a packet when an adequate specified route cannot be found.

When a specified route cannot be found to route a packet (a route which is not a default route), then the ip addresses and interfaces configured through this command are drawn on and the first ip address or interface that provides an active outgoing interface is selected.

Syntax:

```
Route map config>ENTRY n SET IP DEFAULT NEXT-HOP {ip-address | interface} [...ip-
address | ...interface]
```

n	Entry number. The entries do not have to be consecutively numbered, however the numerical order is important as the lowest number is checked first. The range is from 1 to 65535.
ip-address | *interface*	Ip address of the next hop or outgoing interface.

Example:

```
Route map office>ENTRY 1 SET IP DEFAULT NEXT-HOP 192.168.0.1
Route map office>
```

g) ENTRY n SET IP NEXT-HOP

This command determines the ip address or interface to use when routing packets matching the entry.

When the packet matches the entry, it draws on the ip addresses and interface configured through this command. The packet is routed towards the first ip address or interface offering an active outgoing interface.

Syntax:

```
Route map config>ENTRY n SET IP NEXT-HOP {ip-address | interface} [...ip-address |
...interface]
```

n	Entry number. The entries do not have to be consecutively numbered, however the numerical order is important as the lowest number is checked first. The range is from 1 to 65535.
ip-address | *interface*	IP address of the next hop or outgoing interface.

Example:

```
Route map office>ENTRY 1 SET IP NEXT-HOP 192.168.0.1
Route map office>
```

h) ENTRY n SET IP PRECEDENCE

This command induces the packets ip header precedence field matching the entry to establish.

All packets matching this entry will have the precedence field established with the value configured through this command.

Syntax:

```
Route map config>ENTRY n SET IP PRECEDENCE precedence
```

n	Entry number. The entries do not have to be consecutively numbered, however the numerical order is important as the lowest number is checked first. The range is from 1 to 65535.
precedence	Value to establish in the ip header precedence field. You can specify this in numerical form or through a predefined name. The range is from 0 to 7.

The predefined names for the *precedence* field are those shown in the following table:

Number	Name
0	routine
1	priority
2	immediate
3	flash
4	flash-override
5	critical
6	internet
7	network

Example:

```
Route map office>ENTRY 1 SET IP PRECEDENCE PRIORITY
Route map office>
```

i) ENTRY n SET IP TOS

This command induces the packets ip header tos field matching the entry to establish.

All packets matching this entry will have the tos field established with the value configured through this command.

Syntax:

```
Route map config>ENTRY n SET IP TOS service-type
```

n	Entry number. The entries do not have to be consecutively numbered, however the numerical order is important as the lowest number is checked first. The range is from 1 to 65535.
service-type	Value to establish in the ip header tos field. You can specify this in numerical form or through a predefined name. The range is from 0 to 15.

The predefined names for the *type-of-service* field are those shown in the following table:

Number	Name
0	normal
1	min-monetary-cost
2	max-reliability
4	max-throughput
8	min-delay

Example:

```
Route map office>ENTRY 1 SET IP TOS MIN-DELAY
Route map office>
```

j) NO ENTRY n

This command eliminates the specified entry. None of the other entries is affected.

Syntax:

```
Route map config>NO ENTRY n
```

n	Entry number to eliminate. The range is from 1 to 65535.

Example:

```
Route map office>NO ENTRY 1
Route map office>
```

k) NO ENTRY n MATCH IP ADDRESS

This command deletes the specified access lists from the entry check list.

If you do not specify any access list, this command deletes all the access lists from the entry check list.

Syntax:

```
Route map config>NO ENTRY n MATCH IP ADDRESS [...access-list]
```

n	Entry number. The range is from 1 to 65535.
access-list	Access control list number. The range is from 1 to 199.

Example:

```
Route map office>NO ENTRY 1 MATCH IP ADDRESS 101
Route map office>
```

l) NO ENTRY n MATCH LENGTH

This command disables the packet length check in an entry.

Syntax:

```
Route map config>NO ENTRY n MATCH LENGTH
```

n	Entry number. The range is from 1 to 65535.

Example:

```
Route map office>NO ENTRY 1 MATCH LENGTH
Route map office>
```

m) NO ENTRY n SET IP DEFAULT NEXT-HOP

This command eliminates the specified ip addresses and interfaces from the list previously configured through the **entry** *n* **set ip default next-hop** command.

If you do not specify any ip address or interface, the entire list previously configured through the **entry** *n* **set ip default next-hop** command is deleted.

Syntax:

```
Route map config>NO ENTRY n SET IP DEFAULT NEXT-HOP [...ip-address | ...interface]
```

n	Entry number. The range is from 1 to 65535.
ip-address | *interface*	Ip address or interface to delete from the list.

Example:

```
Route map office>NO ENTRY 1 SET IP DEFAULT NEXT-HOP
Route map office>
```

n) NO ENTRY n SET IP NEXT-HOP

This command eliminates the specified ip addresses and interfaces from the list previously configured through the **entry** *n* **set ip next-hop** command.

If you do not specify any ip address or interface, the entire list previously configured through the **entry** *n* **set ip next-hop** command is deleted.

Syntax:

```
Route map config>NO ENTRY n SET IP NEXT-HOP [...ip-address | ...interface]
```

n	Entry number. The range is from 1 to 65535.
ip-address | *interface*	Ip address or interface to delete from the list.

Example:

```
Route map office>NO ENTRY 1 SET IP NEXT-HOP
Route map office>
```

o) NO ENTRY n SET IP PRECEDENCE

This command means that the ip header precedence field for the packets matching the entry is not modified.

Syntax:

```
Route map config>NO ENTRY n SET IP PRECEDENCE
```

n	Entry number. The range is from 1 to 65535.

Example:

```
Route map office>NO ENTRY 1 SET IP PRECEDENCE
Route map office>
```

p) NO ENTRY n SET IP TOS

This command means that the ip header tos field for the packets matching the entry is not modified.

Syntax:

```
Route map config>NO ENTRY n SET IP TOS
```

n	Entry number. The range is from 1 to 65535.

Example:

```
Route map office>NO ENTRY 1 SET IP TOS
Route map office>
```

2.4. EXIT

This command finalizes route map configuring and returns to the route map configuration menu.

Syntax:

```
Route map config>EXIT
```

Example:

```
Route map office>EXIT
Route map config>
```

3. Enabling Policy Routing

Policy Routing is independently enabled in each interface so that all packets entering through an interface are affected by the policy routing configuration for the said interface.

You can also enable Policy Routing for locally generated packets i.e. in the device itself (packets that have not entered through an interface).

In order to configure Policy Routing, access the general configuration menu:

```
*PROCESS 4
Config>
```

From the general configuration menu, access the ip protocol configuration menu:

```
Config>PROTOCOL IP

-- Internet protocol user configuration --
IP config>
```

The commands used to enable Policy Routing are as follows:

Command	Aim
local policy route-map *map-id*	Enables Policy Routing for locally generated packets.
Policy *interface* **route-map** *map-id*	Enables Policy Routing for packets received through an interface.
no local policy route-map	Disables Policy Routing for locally generated packets.
no policy *interface* **route-map**	Disables Policy Routing for packets received through an interface.

To check the policy routing configuration, use the **LIST POLICY** command.

These commands are explained in the following paragraphs.

3.1. LIST POLICY

This command displays the policy routing configuration in those interfaces where this is enabled.

Syntax:

```
IP config>LIST POLICY
```

Example:

```
IP config>LIST POLICY


Ip policy routing:
  Interface      Route map
  ethernet0/0    office
  serial0/0      extern
  local          admin
IP config>
```

3.2. LOCAL POLICY ROUTE-MAP

This command enables Policy Routing for locally generated packets i.e. those packets that have not been received through an interface. This also defines the route map to be used with the said packets.

Syntax:

```
IP config>LOCAL POLICY ROUTE-MAP map-id
```

map-id	Name of the route map to be used.

Example:

```
IP config>LOCAL POLICY ROUTE-MAP OFFICE
IP config>
```

3.3. POLICY interface ROUTE-MAP

This command enables Policy Routing for packets received through the specified interface. This also defines the route map to be used with the said packets.

Syntax:

```
IP config>POLICY interface ROUTE-MAP map-id
```

interface	Interface where Policy Routing is enabled.
map-id	Name of the route map to be used.

Example:

```
IP config> POLICY ethernet0/0 ROUTE-MAP office
IP config>
```

3.4. NO LOCAL POLICY ROUTE-MAP

This command disables Policy Routing for locally generated packets i.e. packets that have not been received through an interface.

Syntax:

```
IP config>NO LOCAL POLICY ROUTE-MAP
```

Example:

```
IP config>NO LOCAL POLICY ROUTE-MAP
IP config>
```

3.5. NO POLICY interface ROUTE-MAP

This command disables Policy Routing for packets received through the specified interface.

Syntax:

```
IP config>NO POLICY interface ROUTE-MAP
```

interface	Interface where Policy Routing is enabled.

Example:

```
IP config>NO POLICY ethernet0/0 ROUTE-MAP
IP config>
```

4. Commands Summary

feature route-map

list

no route-map *map-id*

route-map *map-id*

entry *n* **default**

entry *n* **deny**

entry *n* **match ip address** {*access-list*} [*...access-list*]

entry *n* **match length** *minimum-length maximum-length*

entry *n* **permit**

entry *n* **set ip default next-hop** {*ip-address* | *interface*} [*...ip-address* | *...interface*]

entry *n* **set ip next-hop** {*ip-address* | *interface*} [*...ip-address* | *...interface*]

entry *n* **set ip precedence** *precedence*

entry *n* **set ip tos** *service-type*

no entry *n*

no entry *n* **match ip address** [*...access-list*]

no entry *n* **match length**

no entry *n* **set ip default next-hop** [*...ip-address* | *...interface*]

no entry *n* **set ip next-hop** [*...ip-address* | *...interface*]

no entry *n* **set ip precedence**

no entry *n* **set ip tos**

exit

exit

protocol ip

list policy

local policy route-map *map-id*

policy *interface* **route-map** *map-id*

no local policy route-map

no policy *interface* **route-map**

Chapter 3
Policy Routing Monitoring

1. Monitoring tools

Policy Routing functionality has the following monitoring mechanisms available:

1. Access lists statistics
2. POLR subsystem events.

The access lists statistics used in the route maps provide information on how many packets have matched each access list entry (and therefore with the route map). These also offer information on the last packet that matched each entry.

Example:

```
*PROCESS 3

+FEATURE ACCESS
-- Access Lists user console --
Access Lists>LIST ALL ALL

Standard Access List 1, assigned to Route map

 ACCESS LIST ENTRIES
1     PERMIT  SRC=172.24.51.104/32
        Hits: 277
  (172.24.51.104 <-> 172.24.78.116  Conn:0x0  ICMP  TYPE=8 CODE=0  ECHO  DCSP:0)

Access Lists>
```

You can obtain detailed information on the actions carried out by the Policy Routing subsystem through the POLR subsystem events as explained in the following chapter.

Example:

```
*PROCESS 3

+EVENT

-- ELS Monitor --
ELS>LIST SUBSYSTEM POLR
Event         Level     Message

POLR.001      P-TRACE   tst %I -> %I len %u prt %u local
POLR.002      P-TRACE   tst %I -> %I len %u prt %u int %s
POLR.003      P-TRACE   dis local
POLR.004      P-TRACE   dis int %s
POLR.005      P-TRACE   mis %I -> %I len %u prt %u local rtmap %s
POLR.006      P-TRACE   mis %I -> %I len %u prt %u int %s rtmap %s
POLR.007      P-TRACE   mch %I -> %I len %u prt %u local rtmap %s entry %u
POLR.008      P-TRACE   mch %I -> %I len %u prt %u int %s rtmap %s entry %u
POLR.009      P-TRACE   set %I -> %I tos 0x%02x to 0x%02x
POLR.010      P-TRACE   set %I -> %I nxt hop int %s
POLR.011      P-TRACE   set %I -> %I nxt hop %I
POLR.012      P-TRACE   fwd %I -> %I rt tbl
POLR.013      P-TRACE   set %I -> %I def nxt hop int %s
POLR.014      P-TRACE   set %I -> %I def nxt hop %I
```

```
POLR.015      P-TRACE   def %I -> %I
ELS>ENABLE TRACE SUBSYSTEM POLR ALL
ELS>VIEW
ELS>03/24/03 10:27:27 *POLR.006 mis 172.24.77.253 -> 172.24.255.255 len 78 prt 17 in
t ethernet0/0 rtmap myhost
03/24/03 10:27:27  POLR.008 mch 172.24.51.104 -> 172.24.78.116 len 60 prt 1 int
ethernet0/0 rtmap myhost entry 25
03/24/03 10:27:27  POLR.009 set 172.24.51.104 -> 172.24.78.116 tos 0x00 to 0x10
03/24/03 10:27:27  POLR.012 fwd 172.24.51.104 -> 172.24.78.116 rt tbl
```

Chapter 4
Policy Routing Events

1. Introduction

The POLR subsystem events are described in this chapter. This is associated with the Policy Routing functionality.

To activate the Policy Routing events:

From monitoring:

```
*PROCESS 3

+EVENT

-- ELS Monitor --
ELS>ENABLE TRACE SUBSYSTEM POLR ALL
ELS>
```

From configuration:

```
*PROCESS 4


Config>EVENT

-- ELS Config --
ELS config>ENABLE TRACE SUBSYSTEM POLR ALL
ELS config>
```

So that these are stored in the device configuration, the user must save the configuration and if necessary, restart the device.

2. Events

POLR.001

Level: Per packet trace, P-TRACE

Short Syntax:

POLR.001 tst *srcaddr -> dstaddr* len *length* prt *protocol* local

Long Syntax:

POLR.001 test local packet from *srcaddr* for *dstaddr* of length *length* protocol *protocol*

Description:

A packet was generated from specified address for specified address. Packet length and protocol are also shown. This packet will be tested for policy routing.

Cause:

The packet is being tested for policy routing to find out next hop.

POLR.002

Level: Per packet trace, P-TRACE

Short Syntax:

POLR.002 tst *srcaddr -> dstaddr* len *length* prt *protocol* int *interface*

Long Syntax:

POLR.002 test packet from *srcaddr* for *dstaddr* of length *length* protocol *protocol* from interface *interface*

Description:

A packet was received from specified address for specified address, on specified interface. Packet length and protocol are also shown. This packet will be tested for policy routing.

Cause:

The packet is being tested for policy routing to find out next hop.

POLR.003

Level: Per packet trace, P-TRACE

Short Syntax:

POLR.003 dis local

Long Syntax:

POLR.003 local policy routing disabled

Description:

No route map associated to locally generated packets.

Cause:

Cannot perform policy routing on locally generated packets because there is no associated route map.

Action:

Check configuration of local policy routing and route maps.

POLR.004

Level: Per packet trace, P-TRACE

Short Syntax:

POLR.004 dis int *interface*

Long Syntax:

POLR.004 policy routing disabled on interface *interface*

Description:

No route map associated to specified interface.

Cause:

Cannot perform policy routing on specified interface because there is no associated route map.

Action:

Check configuration of policy routing and route maps for that interface.

POLR.005

Level: Per packet trace, P-TRACE

Short Syntax:

POLR.005 mis *srcaddr -> dstaddr* len *length* prt *protocol* local rtmap *routemap*

Long Syntax:

POLR.005 missed local packet from *srcaddr* for *dstaddr* of length *length* protocol *protocol* with route map *routemap*

Description:

A packet was generated from specified address for specified address. Packet length and protocol are also shown. This packet did not match the specified route map.

Cause:

The packet is not policy routed because it does not match the route map criteria.

POLR.006

Level: Per packet trace, P-TRACE

Short Syntax:

POLR.006 mis *srcaddr -> dstaddr* len *length* prt *protocol* int *interface* rtmap *routemap*

Long Syntax:

POLR.006 missed packet from *srcaddr* for *dstaddr* of length *length* protocol *protocol* interface *interface* with route map *routemap*

Description:

A packet was received from specified address for specified address, on specified interface. Packet length and protocol are also shown. This packet did not match the specified route map.

Cause:

The packet is not policy routed because it does not match the route map criteria.

POLR.007

Level: Per packet trace, P-TRACE

Short Syntax:

POLR.007 mch *srcaddr -> dstaddr* len *length* prt *protocol* local rtmap *routemap* entry *entrynumber*

Long Syntax:

POLR.007 match of local packet from *srcaddr* for *dstaddr* of length *length* protocol *protocol* with route map *routemap* entry *entrynumber*

Description:

A packet was generated from specified address for specified address. Packet length and protocol are also shown. This packet did match the specified entry of the specified route map.

Cause:

The packet is policy routed because it does match the route map criteria.

POLR.008

Level: Per packet trace, P-TRACE

Short Syntax:

POLR.008 mch *srcaddr* -> *dstaddr* len *length* prt *protocol* int *interface* rtmap *routemap* entry *entrynumber*

Long Syntax:

POLR.008 match of packet from *srcaddr* for *dstaddr* of length *length* protocol *protocol* interface *interface* with route map *routemap* entry *entrynumber*

Description:

A packet was received from specified address for specified address, on specified interface. Packet length and protocol are also shown. This packet did match the specified entry of the specified route map.

Cause:

The packet is policy routed because it does match the route map criteria.

POLR.009

Level: Per packet trace, P-TRACE

Short Syntax:

POLR.009 set *srcaddr* -> *dstaddr* tos 0x*oldtos* to 0x*newtos*

Long Syntax:

POLR.009 set packet from *srcaddr* for *dstaddr* field tos changed from 0x*oldtos* to 0x*newtos*

Description:

A packet was received from specified address for specified address. The tos field was changed from oldtos to newtos.

Cause:

The packet is policy routed setting a new value for its tos field.

POLR.010

Level: Per packet trace, P-TRACE

Short Syntax:

POLR.010 set *srcaddr* -> *dstaddr* nxt hop int *interface*

Long Syntax:

POLR.010 set packet from *srcaddr* for *dstaddr* next hop interface *interface*

Description:

A packet was received from specified address for specified address. The next hop is set to the specified interface.

Cause:

The packet is policy routed setting the output interface.

POLR.011

Level: Per packet trace, P-TRACE

Short Syntax:

POLR.011 set *srcaddr* -> *dstaddr* nxt hop *nexthop*

Long Syntax:

POLR.011 set packet from *srcaddr* for *dstaddr* next hop *nexthop*

Description:

A packet was received from specified address for specified address. The next hop is set to the specified address.

Cause:

The packet is policy routed setting the next hop to the specified address.

POLR.012

Level: Per packet trace, P-TRACE

Short Syntax:

POLR.012 fwd *srcaddr -> dstaddr* rt tbl

Long Syntax:

POLR.012 forward packet from *srcaddr* for *dstaddr* with routing table

Description:

A packet was received from specified address for specified address. The packet is forwarded based on routing table rather than policy routing.

Cause:

The packet did match a route map, but anyway it is being forwarded by means of routing table because no next hop set clause was active.

POLR.013

Level: Per packet trace, P-TRACE

Short Syntax:

POLR.013 set *srcaddr -> dstaddr* def nxt hop int *interface*

Long Syntax:

POLR.013 set packet from *srcaddr* for *dstaddr* default next hop interface *interface*

Description:

A packet was received from specified address for specified address. The default next hop is set to the specified interface.

Cause:

The packet is policy routed setting the default output interface.

POLR.014

Level: Per packet trace, P-TRACE

Short Syntax:

POLR.014 set *srcaddr -> dstaddr* def nxt hop *nexthop*

Long Syntax:

POLR.014 set packet from *srcaddr* for *dstaddr* default next hop *nexthop*

Description:

A packet was received from specified address for specified address. The default next hop is set to the specified address.

Cause:

The packet is policy routed setting the default next hop to the specified address.

POLR.015

Level: Per packet trace, P-TRACE

Short Syntax:

POLR.015 def *srcaddr -> dstaddr*

Long Syntax:

POLR.015 default routing for packet from *srcaddr* for *dstaddr*

Description:

A packet was received from specified address for specified address. No specific route was found for this packet, therefore default routing is performed.

Cause:

The packet is not policy routed because no specific route was found.