QoS Policy Propagation with BGP (QPPB)

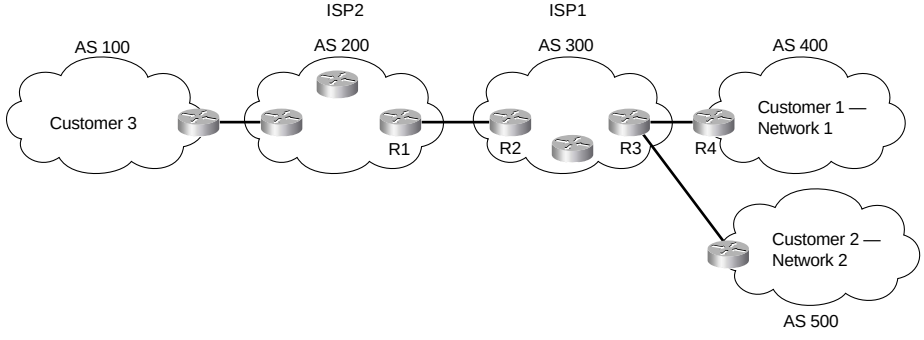
QoS policies that differentiate between different types of traffic can be most easily defined for a single enterprise network. For instance, one enterprise may want to treat important web traffic, not-important web traffic, and all other data traffic as three different classes, and use different classes for voice and video traffic. For the Internet, however, a single QoS policy would never work. Differentiated services (DiffServ), which was designed specifically to address QoS over the Internet, defines the role of ingress boundary nodes to re-mark traffic as it enters a different DiffServ domain, essentially changing the differentiated services code point (DSCP) to reflect the QoS policies of each respective DiffServ domain. This practice allows each DiffServ domain to set its own QoS policies.

QoS policies that classify traffic based on the characteristics of the flow—voice, video, different data applications, and so on—can be defined and used in enterprises and by service providers. Enterprises can afford to be more selective, because a single group can often set the QoS policies. For instance, an enterprise could classify based on the IP addresses of some mission- critical servers. QoS policies for Internet service providers (ISPs) tend to be less specific than those for an enterprise, because ISPs have many customers. However, ISPs can still implement QoS policies based on the type of traffic contained in the packet.

ISPs may want a QoS policy just to prefer one customer's traffic over another. In Figure B-9, for instance, consider ISP 1, which has two customers. Customer 1 has agreed to pay a premium for its Internet service, in return for ISP 1 agreeing to provide better latency and delay characteristics for the traffic. Customer 2 keeps paying the same amount as always, and still gets best- effort service.

The QoS tools only need to differentiate between Customer 1 and Customer 2 traffic to support this policy. So, for packets flowing from right to left, if the source IP address is an IP address in Customer 1's network, the packet might be marked with precedence 4, for instance. Similarly, when packets flow left to right, these same tools could examine the destination IP address, and if it's part of Customer 1's network, precedence 4 could be marked. Packets to or from Customer 2 could be marked with precedence 0.

Figure B-9 *QoS Policy Based on Customer—Customer 1 and Customer 2*



Class-based (CB) marking, policy-based routing (PBR), and committed access rate (CAR) could perform the necessary marking to support premium and best-effort customer services. However, each of these three tools has some negative side effects. For all three tools, that classification would require an IP ACL for matching the packets, for all packets. For an ISP with many customers, however, classifying and marking packets based on referencing ACLs for a large number of packets may induce too much overhead traffic. Suppose further that ISP 1 and ISP 2 agree to support each other's premium and best-effort customers in a similar manner. The two ISP's would have to continually exchange information about which networks are premium, and which are not, if they are using IP ACLs to classify the traffic. Additionally, when new customers are added, ISP 1 may be waiting on ISP 2 to update their QoS configuration before the desired level of service is offered to the new customer.

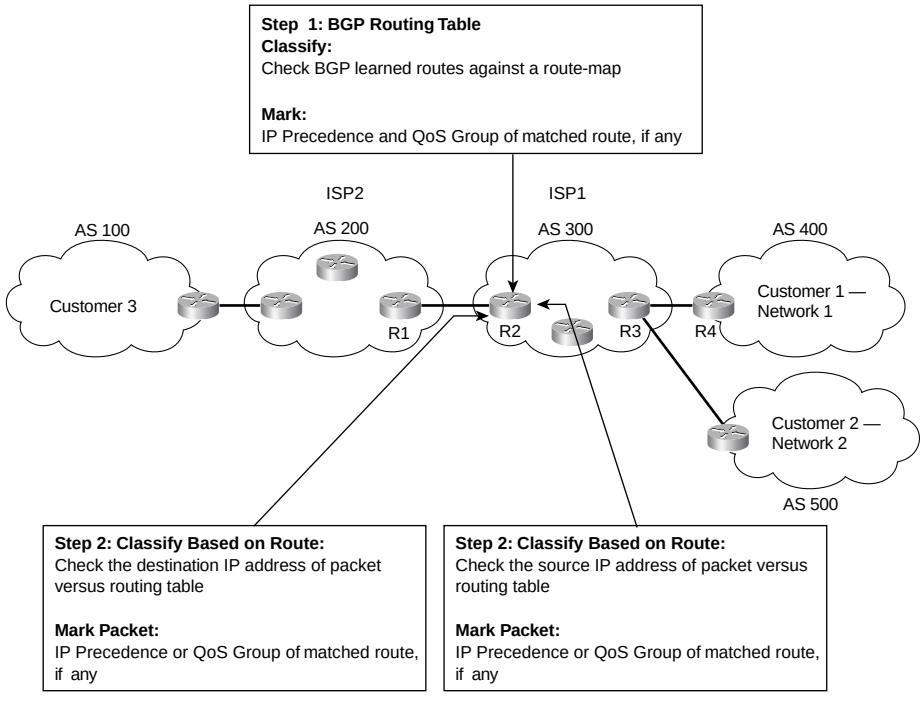
To overcome the two issues—the scalability of classifying based on ACLs, and the administrative problems of just listing the networks that need premium services—QPPB was created. QPPB allows marking of packets based on an IP precedence or QoS group value associated with a Border Gateway Protocol (BGP) route. For instance, the BGP route for Customer 1's network, Network A, could be given a BGP path attribute that both ISP 1 and ISP 2 agree should mean that this network

receives better QoS service. Because BGP already advertises the routes, and the QoS policy is based on the networks described in the routes, QPPB marking can be done more efficiently than with the other classification and marking tools.

Figure B-10 shows the basic process in action. In this example, R3 is configured to use QPPB, although it would likely be used in several places around the network.

QPPB follows two steps: marking routes, and then marking packets based on the values marked on the routing entries. BGP routing information includes the network numbers used by the various customers, and other BGP path attributes. Because Cisco has worked hard over the years to streamline the process of table lookup in the routing table, to reduce per-packet processing for the forwarding process, QPPB can use this same efficient table-lookup process to reduce classification and marking overhead.

Figure B-10 *QPPB—Basic Components*



For reference, Tables B-8 and B-9 summarize the QPPB configuration and exec commands, respectively.

Table B-8 *Configuration Command Reference for QPPB*

Command	Mode and Function
route-map *map-tag* [**permit** | **deny**] [*sequence-number*]	Global command; creates a route map entry
match ip address {*access-list-number* | *access-list-name*} [... *access-list-number* | ... *access-list-name*]	Route-map subcommand; used to match IP packets based on parameters matchable with an IP ACL
match length *minimum-length maximum-length*	Route-map subcommand; used to mach IP packets based on their length
set ip precedence *number* | *name*	Route-map subcommand; sets IP precedence vale using the decimal number of name.
set ip qos-group group-id	Route-map subcommand; sets a group ID in the routing table for classification throughout the network.
table-map *map-name*	BGP subcommand; used to modify values related to BGP learned routes, including precedence and QoS group
ip community-list *community-list-number* {**permit** | **deny**} *community-number*	Global command; used to create a community list, which matches values in the BGP community string
ip as-path access-list *access-list-number* {**permit** | **deny**} *as-regexp*	Global command; used to create an autonomous system (AS) path list, which matches values in the autonomous system number (ASN) path BGP attribute
ip bgp-community new-format	BGP subcommand; used to make IOS use the *AA*:*NN* format for community values, with *AA* being the ASN, and *NN* being a user-defined value
bgp-policy ip-prec-map	Interface subcommand; enables QPPB for packets entering the interface, marking IP precedence
bgp-policy ip-qos-map	Interface subcommand; enables QPPB for packets entering the interface, marking QoS group

Table B-9 *EXEC Command Reference for QPPB*

Command	Function
show ip bgp	Shows BGP routing table
show ip route *prefix*	Shows IP routing table entries, including precedence values
show ip bgp community-list *community-list-number*	Lists configuration of the community list
show ip cef *network*	Shows the Cisco Express Forwarding (CEF) Forwarding Information Base (FIB), including the marked QoS values

QPPB can be a confusing topic. The rest of this section discusses more detail about how QPPB works and how to configure it. One key to understanding QPPB, in spite of some of the detail, is to keep these two key points in mind as you read the following sections:

- QPPB classifies BGP routes based on the BGP routes' attributes, and marks BPG routes with an IP precedence or QoS group value.
- QPPB classifies packets based on the associated routing table entries, and marks the packets based on the marked values in the routing table entry.

Because QPPB involves quite a few detailed concepts and configuration, some of the true details of how QPPB works are glossed over during the initial discussions. These details are explained at the end of this section in the subsection titled "QPPB: The Hidden Details."

QPPB Route Marking: Step 1

QPPB allows routers to mark packets based on information contained in the routing table. Before packets can be marked, QPPB first must somehow associate a particular marked valued with a particular route. QPPB, as the name implies, accomplishes this task using BGP. This first step can almost be considered as a separate classification and marking step by itself, because BGP routes are classified, based on information that describes the route, and marked with some QoS value.

The classification feature of QPPB can examine many of the BGP path attributes. The two most useful BGP attributes for QPPB are the autonomous system number (ASN) sequence, referred to as the autonomous system path, and the community string. The autonomous system path contains the ordered list of ASNs, representing the ASNs between a router and the autonomous system of the network described in the route. In Figure B-10, R1 receives a BGP update for Network 1, listing ASNs 300 and 400 in the autonomous system path and a BGP update for Network 2, listing ASNs 300 and 500 in the autonomous system path. QPPB can be used to mark the route to Network 1 (Customer 1) with one precedence value, while marking the route to Network 2 (Customer 2) with

another precedence value, based on the autonomous system path received for the route to each customer.

The community attribute provides a little more control than does the autonomous system path. The autonomous system path is used to avoid routing loops, and the contents of the autonomous system path changes when aggregate routes are formed. The community attribute, however, allows the engineer to essentially mark any valid value. For instance, R3 could set the community attribute to 10:200 for the route to Network 1, and advertise that route toward the left side of the network diagram. Other routers could then use QPPB to classify based on the community attribute of 10:200, and assign the appropriate precedence value to the route to Network 1. QPPB configuration would essentially create logic as follows: "If the community attribute contains 10:200, mark the route with precedence 4."

Example B-10 lists the QPPB configuration just for marking the route based on the autonomous system number. With this configuration, no packets are marked, because the QPPB configuration is not complete. (The complete configuration appears in the next section.) QPPB is a two- step process, and Example B-1 just shows the configuration for the first step.

Example B-10 *QPPB Route Marking with BGP Table Map: R2*

```
router bgp 300
 table-map mark-prec4-as400
!
route-map mark-prec4-as400 10
 match as-path 1
 set ip precedence 4
!
route-map mark-prec4-as400 20
 set ip precedence 0
!
ip as-path access-list 1 permit _400_
```

This example shows R2's configuration for QPPB. (Note that the entire BGP configuration is not shown, just the configuration pertinent to QPPB.) The **table-map** BGP router subcommand tells IOS that, before adding BGP routes to the routing table, it should examine a route map called mark-prec4-as400. Based on the **match** and **set** commands in the route map, when BGP adds routes to the routing table, it also associates either precedence 4 or precedence 0 with each route.

The route map has two clauses—one that matches routes that have autonomous system 400 anywhere in the autonomous system path sequence attribute, and a second clause that matches all routes. Clause 10 matches ASN 400 by referring to autonomous system path ACL 1, which matches any autonomous system path containing ASN 400, and sets the precedence to 4 for those routes. Clause 20 matches all packets, because no specific **match** command is configured, and sets the precedence to 0.

QPPB Per-Packet Marking: Step 2

After QPPB has marked routes with IP precedence or QoS group values, the packet marking part must be performed. After the packets have been marked, traditional QoS tools can be used to perform queuing, congestion avoidance, policing, and so on, based on the marked value.

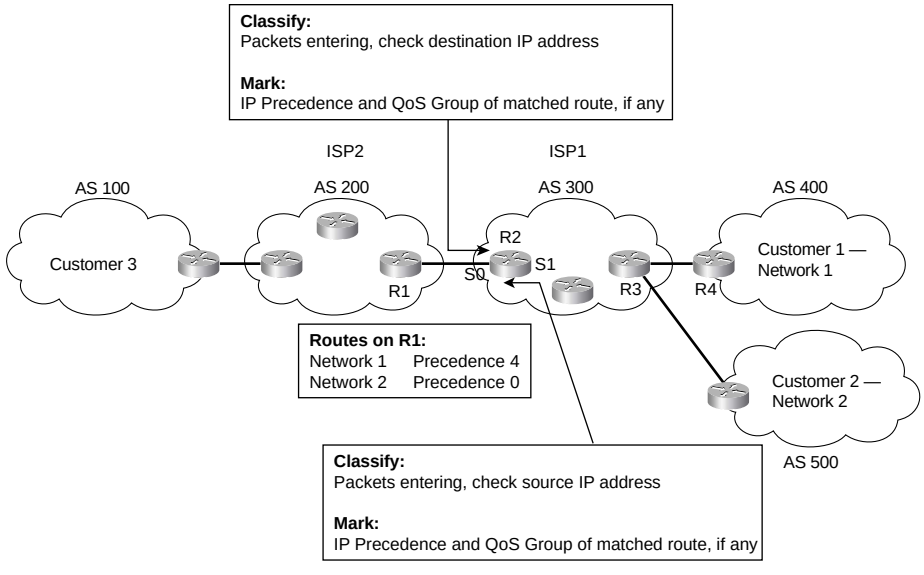
QPPB's packet-marking logic flows as follows:

1. Process packets entering an interface.
2. Match the destination or source IP address of the packet to the routing table.
3. Mark the packet with the precedence or QoS group value shown in the routing table entry.

The three-step logic for QPPB packet marking follows the same general flow as the other classification and marking tools; in this case, however, the classification options, and the marking options, are quite limited. QPPB packet classification is based on the routing table entry that matches the packet, and QPPB packet marking just marks the packet with the same value found marked in the route.

Figure B-11 shows with the same network, but with the marking logic on R2 shown.

Figure B-11 *QPPB Per-Packet Marking Logic*



QPPB allows for marking of packets that have been sent to Customer 1, and for marking packets that have been sent by Customer 1. For packets entering R2's S0 interface, for instance, the packet is going toward Customer 1, so the destination IP address is in Network 1. Therefore, the QPPB

logic on R2's S0 should compare the packet's destination IP address to the routing table; if the appropriate QoS field has been set in the route, the packet is marked with the same value. That takes care of packets passing through R3 that are headed to Customer 1.

For packets that Customer 1 has sent, going from right to left in the figure, QPPB on R2 can still mark the packets. These packets typically enter R2's S1 interface, however, and the packets have a *source* IP addresses in Network 1. To associate these packets with Network 1, QPPB examines the routing table entry that matches the packet's source IP address. This match of the routing table is not used for packet forwarding; it is used only for finding the precedence or the QoS group value to set on the packet. In fact, the table lookup for destination addresses does not replace the normal table lookup for forwarding the packet, either. Because the routing table entry for Network 1 has IP precedence set to 4, QPPB marks these packets with precedence 4.

Example B-11 shows the completed configuration on R2, with the additional configuration for per-packet marking highlighted.

Example B-11 *QPPB: Completed Example on R2*

```
ip cef
!
Router bgp 300
 table-map mark-prec4-as400
!
route-map mark-prec4-as400 10
 match as-path 1
 set ip precedence 4
!
route-map mark-prec4-as400 20
 set ip precedence 0
!
ip as-path access-list 1 permit _400_
!
interface Serial0
 bgp-policy destination ip-prec-map
!
interface serial1
 bgp-policy source ip-prec-map
```

The **bgp-policy** interface subcommand enables QPPB for packets entering the interface. The **destination** or **source** keyword identifies whether QPPB should perform table lookup on the packets' destination or source addresses, respectively. On S0, the **destination** keyword is used, because the packets entering S0 presumably are going toward Customer 1. Conversely, on S1 the **source** keyword is used, because the packets entering S1 presumably were sent by Customer 1. Finally, the **ip-prec-map** keyword implies that the precedence should be set based on the routing table entry, and not the QoS group.

QPPB Sample Configuration

QPPB can classify based on both the autonomous system path and the community string. BGP considers the autonomous system path as a well-known mandatory path attribute; therefore, in the earlier examples, R3 could just examine the autonomous system path. Conversely, BGP considers the community string to be an optional transitive attribute—which means that the community string does not have to be set, and is not set without some additional configuration causing it to be set.

Example B-12 shows the same network, with the same goal of giving Customer 1 premium service. In this example, however, the BGP community attribute is used. The community attribute is set by R3, for routes received from Customer 1 via BGP. With the community attribute set, other routers can use it for classifying the BGP routes and marking the routes with precedence 4. The example shows R3's QPPB configuration. Example B-12 lists the configuration on R3, and Example B-13 lists the configuration on R2.

Example B-12 *QPPB Sample Based on BGP Community: R3 Configuration*

```
router bgp 300
 neighbor 192.168.1.1 remote-as 400
 neighbor 192.168.1.1 route-map set-comm in
 neighbor 192.168.2.2 remote-as 300
 neighbor 192.168.2.2 send-community
!
route-map set-comm permit 10
 set community 4:50
```

Example B-13 *QPPB Sample Based on BGP Community: R2 Configuration*

```
ip cef
!
router bgp 300
 table-map mark-prec4-comm
!
route-map mark-prec4-comm permit 10
 match community 1
 set ip precedence 4
!
route-map mark-prec4-comm permit 20
 set ip precedence 0
!
ip community-list 1 permit 4:50
!
interface Serial0
 bgp-policy destination ip-prec-map
!
interface serial1
 bgp-policy source ip-prec-map
```

In Example B-12, R3 has just set the community string to 4:50 for BGP routes learned from neighbor 192.168.1.1, which is a router at Customer 1. To set the community, BGP uses **route- map set-comm** based on the **neighbor 192.168.1.1 route-map set-comm in** command. This route map contains 1 clause, which matches all routes because there is no **match** command in clause 10, and sets the community string to 4:50. IOS BGP does not forward the community attribute by default, so the **neighbor 192.168.2.2 send-community** command is needed to make R3 send the community string to R2, whose BGP ID is 192.168.2.2. So, R3 has set all incoming routes from R4 with community 4:50, and includes the community attribute in the updates sent to R2.

Example B-13 shows the configuration for QPPB on R2. The configuration is similar to Example B-11, with the highlighted sections pointing out the added or changed configuration. The **table-map** BGP router subcommand still directs BGP to mark the routes with precedence 4, but this time using a new route map, mark-prec4-comm. This route map uses two clauses. The first clause, clause 10, matches the community set by R3 by referring to IP community list 1 using the **match community 1** command. The community list, created in the single global command **ip community-list 1 permit 4:50**, just matches all BGP routes whose community string contains 4:50. Route map mark-prec4-comm sets IP precedence 4 for BGP routes that match the community sting. The second route map clause, clause 20, matches all routes because no explicit **match** statement is configured, and sets the IP precedence to 0 for these routes.

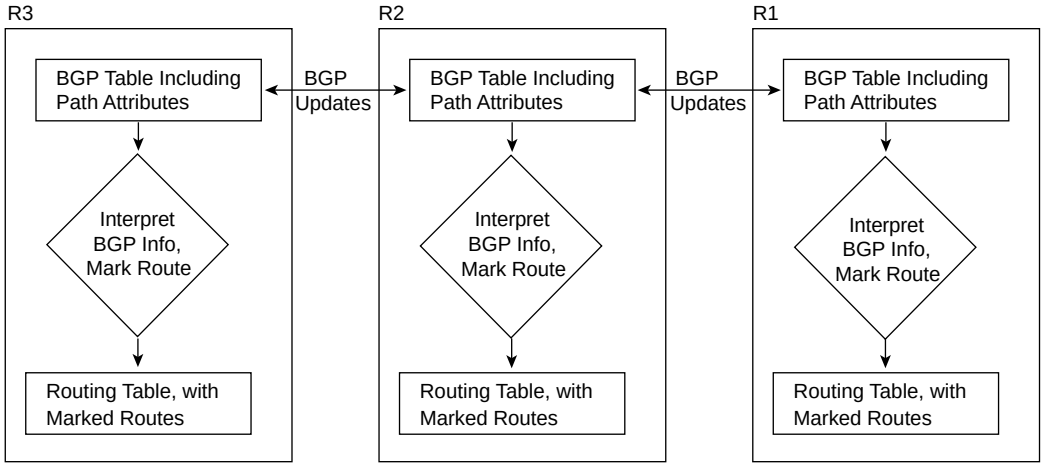
The packet-marking function, as opposed to the route-marking function, is enabled by the **bgp-policy interface** subcommands, which are exactly the same as shown earlier in Example B-6.

QPPB: The Hidden Details

As mentioned earlier, QPPB confuses most people the first time they learn about it. Therefore, you should understand a bit more about it. The first aspect of QPPB you should understand pertains to what BGP updates contain in support of QPPB, and the second aspect of QPPB you should understand is what really happens when QPPB marks a route.

First, BGP updates do *not* include the IP precedence or QoS group value inside the BGP update. QPPB reacts to the information in a normal BGP update to perform QoS marking of BGP routes, and then in turn performs packet marking based on the marked routes. In other words, BGP RFCs did not add any specification for adding a QoS marking field to the information inside the update. Therefore, to mark based on BGP routes, QPPB uses preexisting fields in the BGP update, such as the autonomous system path and the community attribute. In fact, the BGP-4 RFCs added the community attribute to provide a flexible field for marking BGP routes for future unforeseen purposes, such as QPPB. Figure B-12 depicts the general idea:

Figure B-12 *BGP Updates and QPPB Route Marking: No QoS-Marked Fields in BGP Update*



When marking IP precedence in packets, QPPB marks the same field already covered in depth in this chapter—the first 3 bits of the ToS byte. When QPPB marks the QoS group, it actually marks a header that is added to the packet when passing through a 7500, GSR, or ESR series router. However, QPPB must mark the route first, and then mark the packet based on the route that matches the source or destination IP address in the packet. To understand what mark the route really means, you must take at least a cursory look at Cisco Express Forwarding (CEF).

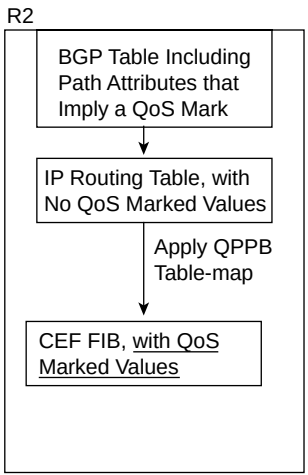
IOS provides several different processing paths in software for forwarding packets. Process switching is one of those paths, and is the most processor-intensive path. Fast switching is another switching path still in use today. CEF is yet another switching or forwarding path, and CEF has been designed to be very efficient. Other switching paths have also been added over the years, some specific to particular hardware models. The one thing all these optimized forwarding paths have in common is that they optimize for the forwarding process by streamlining two functions: the process of matching the correct route in the routing table, and the process of building and adding the new data-link header to the packet.

CEF optimizes forwarding by creating a new table that includes entries for the routes in the routing table. This table is called the *Forwarding Information Base* (FIB). The FIB optimizes the process of locating a route by performing a table lookup in the FIB rather than the less- efficient table lookup of the routing table. In other words, CEF switching crunches the routing table into the FIB, and then uses the FIB to make the forwarding decisions. (This in itself is somewhat of an oversimplification of CEF; for more detail, refer to Vijay Bollapragada's *Inside Cisco IOS Software Architecture* [Cisco Press, 2000].)

CEF optimizes the creation of new data-link headers by creating a table that contains the new data-link header associated with each next-hop IP address in the FIB. By doing so, when FIB table lookup is complete, the header can be added to the packet with little processing.

When QPPB marks a route, it actually marks either or both of the two fields inside each entry in the FIB. The FIB contains IP precedence and QoS group fields in order to support QPPB. Therefore, when CEF crunches the routing table to create FIB entries, when QPPB is configured, the appropriate FIB precedence and QoS group fields are set. Figure B-13 shows the general idea.

Figure B-13 *"Marking the Route": Marking the CEF FIB*



QPPB Summary

QPPB provides convenient classification and marking when BGP is already in use. Because QPPB bases classification decisions on BGP information, however, the classification process should consume less overhead per packet than the other generalized classification and marking tools.