Using Multipath Unicast RPF Check in Supervisor Engine 720

Unicast Reverse Path Forwarding (uRPF) check helps to mitigate problems caused by forged (spoofed) or malformed IP source addresses that are passing through a network device by discarding IP packets that lack a verifiable IP source address. For example, a number of common types of denial-of-service (DoS) attacks, including Smurf, Mstream, and Tribe Flood Network, take advantage of forged or rapidly changing source IP addresses to allow attackers to thwart efforts to locate or filter the attacks.

For Internet Service Providers (ISPs) that provide public access, as well as for small, medium, and large Enterprise networks, uRPF check deflects such attacks by forwarding only packets that have source addresses that are valid and consistent with the IP routing table. This action protects not only the network of the ISP or Enterprise, but the rest of the Internet as well.

Figure 1: Supervisor Engine 720 with Integrated 720 Gbps Switch Fabric

With the introduction of the Supervisor Engine 720 (see Figure 1) on the Cisco Catalyst 6500 Series switches and the Cisco 7600 Series routers, uRPF check is performed in hardware with no performance penalty, even when there are multiple return paths back to the source IP address prefixes in the routing table. With strict-method uRPF check, Supervisor Engine 720 supports two parallel paths for all prefixes in the routing table, and up to six parallel paths for prefixes reached through any of four user-configurable RPF interface groups. With loose-method uRPF check (also known as exist-only method), Supervisor Engine 720 supports up to eight reverse-path interfaces (the Cisco IOS software is limited to eight reverse paths in the routing table).

Note: An IP prefix is simply another name for a route in the IP routing table. Such a prefix might be a host entry (172.16.1.1/32), a subnet or supernet entry (172.16.1.0/24), or even the default route (0.0.0.0/0). The Supervisor Engine 720 performs a longest-match IP prefix lookup to make a forwarding decision or verify a source IP address.

Overview of Unicast RPF Check on Supervisor Engine 720

When unicast RPF check is enabled on a router interface, the system examines all packets received on that interface to make sure that the source IP addresses contained in the packets are valid. This check consists of verifying that the source IP prefix exists in the routing table, and may additionally consist of ensuring that the packet was received on one of the interfaces used to reach that IP prefix (known as a reverse-path interface for the prefix).

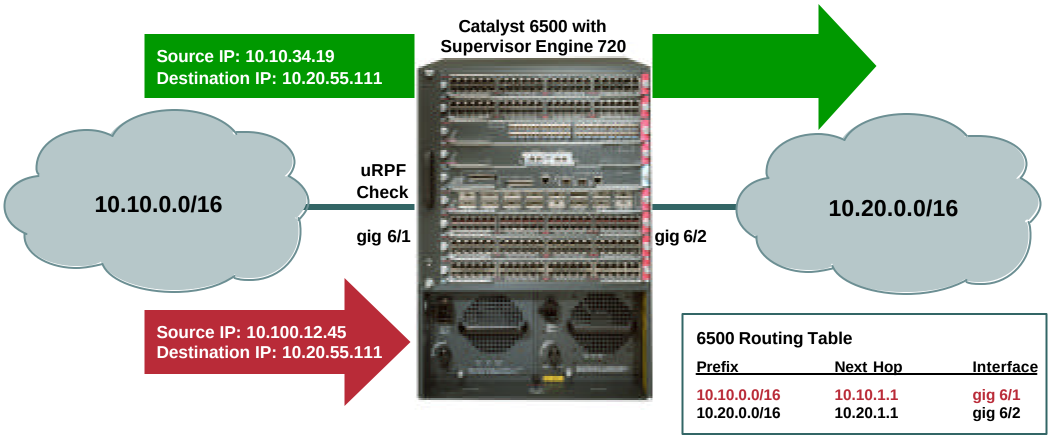
For each packet received on the interface, the system performs a reverse lookup in the routing table. If the packet passes the reverse lookup, the packet is forwarded as normal (assuming there is no additional configuration, such as an output ACL, that results in the packet being dropped). If the packet does not pass the reverse lookup, the packet is discarded.

Note: ACL-based unicast RPF check is not supported in the hardware on Supervisor Engine 720. If you configure ACL-based uRPF check on an interface, all packets denied in the ACL are redirected to the MSFC3 CPU for the uRPF check, while packets permitted by the ACL are forwarded in hardware without a uRPF check. Redirected packets are subject to the global ICMP-unreachable hardware-to-CPU rate limiter (that limits the amount of such traffic that reaches the MSFC3 CPU) and may be dropped in hardware.

Basic uRPF Check Example

Consider the topology shown in Figure 2.

Figure 2: Unicast RPF Example



Prefix	Next Hop	Interface
10.10.0.0/16	10.10.1.1	gig 6/1
10.20.0.0/16	10.20.1.1	gig 6/2

Strict unicast RPF check is enabled on interface GigabitEthernet 6/1, and there are two prefixes (destination networks) in the routing table on the Catalyst 6500 switch: 10.10.0.0/16 and 10.20.0.0/16. Strict uRPF check verifies for each input packet both that the source IP prefix exists, and that the prefix is reached through the input interface. Strict uRPF check is described in detail in the section “Global uRPF Check Methods” on page 4.

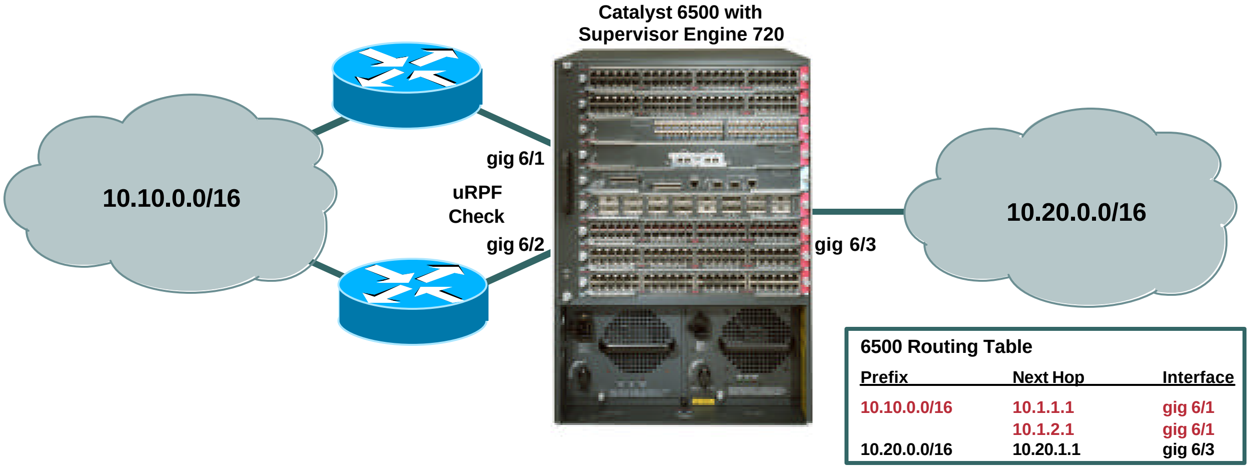
Now suppose a packet arrives on interface GigabitEthernet 6/1 with source IP address 10.10.100.100 and destination IP address 10.20.100.100. The hardware-forwarding engine (the PFC3 on the Supervisor Engine 720 or the DFC3 on a dCEF module) performs a routing table lookup on the destination IP address and an RPF lookup on the source IP address of the packet. This packet passes the uRPF check because, according to the routing table, IP address 10.10.100.100 is reached through interface GigabitEthernet 6/1.

Now suppose a second packet arrives on interface GigabitEthernet 6/1 with source IP address 10.99.100.100 and destination IP address 10.20.100.100. Again the forwarding engine performs both a routing table lookup and an RPF lookup. However, this time, the packet fails the uRPF check and is dropped because a prefix for the source IP address, 10.99.100.100, does *not* exist in the routing table. Without uRPF check, this packet would have been forwarded normally because the destination IP prefix does exist.

Multipath uRPF Check Example

In many networks, there are prefixes that are reachable through more than a single interface. For example, consider the topology shown in Figure 3.

Figure 3: Multipath Unicast RPF Example



6500 Routing Table

Prefix	Next Hop	Interface
10.10.0.0/16	10.1.1.1	gig 6/1
	10.1.2.1	gig 6/1
10.20.0.0/16	10.20.1.1	gig 6/3

In this scenario, the 10.10.0.0/16 network is reached through two different interfaces, GigabitEthernet 6/1 and GigabitEthernet 6/2. Therefore, when unicast RPF check is enabled on both of these interfaces, packets sourced from the 10.10.0.0/16 prefix will pass the uRPF check on either interface.

URPF Check on Supervisor Engine 720

While the Catalyst 6500 Series Supervisor Engine 2 provides uRPF check in hardware on a single interface per prefix, in Supervisor Engine 720, multipath prefixes with up to six reverse paths are supported in hardware. The Supervisor Engine 720 supports two next -hop interfaces per prefix for all prefixes in the routing table. Network administrators can additionally configure up to four uRPF multipath interface groups on Supervisor Engine 720. By using uRPF multipath interface groups, uRPF check for up to six parallel reverse paths per prefix is supported in hardware.

In addition, the Supervisor Engine 720 doubles the number of destination prefix entries available in the system with uRPF check enabled. On Supervisor Engine 2, 128K entries are available in the hardware when uRPF check is enabled. However, regardless of whether uRPF check is enabled, Supervisor Engine 720 always supports the full 256K prefix entries in the hardware-forwarding table.

The unicast RPF check implementation in Supervisor Engine 720 consists of several components that combine to provide uRPF check support in hardware:

- Global uRPF check method
- Global multipath uRPF check mode
- Per-prefix source IP FIB table entries
- Global uRPF check hardware adjacency
- Global uRPF check interface-group table
- Global uRPF hardware-to-CPU rate limiter

- uRPF check CLI and statistics

These different components are described in detail in the following sections.

Global uRPF Check Methods

There are four methods of performing the uRPF check in Cisco IOS:

- Strict uRPF check
- Strict uRPF check with allow default
- Loose uRPF check
- Loose uRPF check with allow default

uRPF check is configured on a per-interface basis. However, although Cisco IOS platforms that forward traffic in software can apply different uRPF check methods on different interfaces, **on Supervisor Engine 720, the uRPF method configured is globally applied to all interfaces in the system that have uRPF check enabled**. Hardware support for different uRPF methods per interface is not implemented. Therefore, when you configure an interface to use a uRPF method that is different from the current global method, all other interfaces in the system that have uRPF check enabled use the new method.

Strict uRPF Check

The strict uRPF check method verifies that traffic received by the system is sourced from a prefix that exists in the routing table, and that the prefix is reachable through the input interface. Packets that do not meet these criteria are discarded. Strict uRPF check provides the greatest level of protection against spoofed or invalid source IP addresses.

Strict uRPF Check with Allow Default

The strict uRPF check with allow default method verifies that, if traffic received by the system is sourced from a prefix that exists in the routing table, the prefix is reachable through the input interface. In addition, if packets sourced from a prefix that does not exist in the routing table are received and a valid default route exists, the packets pass the uRPF check provided they are received on one of the reverse-path interfaces for the default route. If there is no default route present, this mode behaves the same as the strict uRPF check method.

Loose uRPF Check

Also known as exist-only checking, the loose uRPF check method verifies only that traffic received by the system is sourced from a prefix that exists in the routing table, regardless of the interface on which the traffic arrives. Packets that do not meet this criterion are discarded. Loose uRPF check provides a lower level of protection against spoofed or invalid source IP addresses but is useful in asymmetric routing environments where valid traffic might arrive on interfaces that are not reverse-path interfaces for the source prefix in the routing table.

Loose uRPF Check with Allow Default

As long as a default route exists, the behavior of the loose uRPF check with allow default method is the same as not having uRPF check enabled at all—all traffic passes the uRPF check. However, if no default route exists in the routing table, the behavior is the same as loose uRPF check—if the source prefix does not exist, the traffic is discarded.

Global Multipath uRPF Check Modes

There are three modes of operation for multipath unicast RPF in Supervisor Engine 720:

- “Punt” mode—uRPF check is enforced in hardware for up to two interfaces per prefix. Packets arriving on any additional interfaces are redirected (punted) to the MSFC3 CPU for uRPF check in software. **This is the default mode.**
- “Pass” mode—uRPF check is enforced in hardware for single-path and two-path prefixes. uRPF check is disabled for packets coming from multipath prefixes with three or more reverse-path interfaces (these packets always pass the uRPF check).
- “Interface Group” mode—uRPF check is enforced in hardware for single-path and two-path prefixes. uRPF check is also enforced for up to four additional interfaces per prefix through user-configured multipath uRPF check interface groups; uRPF check is disabled for packets coming from other multipath prefixes that have three or more reverse-path interfaces (these packets always pass the uRPF check).

The following sections discuss the behavior of the system when operating in the different modes in more detail.

Punt Mode

In punt mode, Supervisor Engine 720 supports two RPF interfaces for every prefix in the routing table in hardware when using the strict uRPF check method. When using the loose uRPF check method, up to eight RPF interfaces per prefix are supported. Punt mode is the default uRPF check mode in Supervisor Engine 720.

Punt mode and strict uRPF check

When using the strict uRPF check method, if a prefix can be reached through more than two different interfaces, some traffic will be “punted,” or redirected, to the MSFC3 CPU for processing in software. This can have a potentially significant impact on the system, manifested by one of the following symptoms:

- The MSFC3 CPU utilization will increase in direct proportion to the amount of redirected traffic, OR
- Interface throughput will decrease for interfaces on which uRPF check cannot be enforced in hardware (as a result of either the reduced forwarding capability of the software path, the dropping of packets in hardware by the uRPF check hardware-to-CPU rate limiter, or both)

However, this mode was chosen as the default because, unlike pass mode, all packets that are forwarded by the system are subjected to the uRPF check.

In uRPF check punt mode, up to two interfaces per prefix are selected for programming in the hardware (these are the first two interfaces in the group of reverse-path interfaces installed in the routing table). The system performs the uRPF check on traffic entering either of these two interfaces in hardware. Traffic that passes the uRPF check on either of these two interfaces is forwarded in hardware. However, traffic entering any additional interfaces that are valid reverse-path interfaces for the IP prefix will use the global uRPF-check hardware adjacency and are redirected to the MSFC3 CPU for software processing.

There are two scenarios to consider for a punt mode configuration with the strict uRPF check method:

- Scenario 1—All prefixes have less than three valid reverse-path interfaces in the routing table
- Scenario 2—One or more prefixes have three or more valid reverse-path interfaces in the routing table

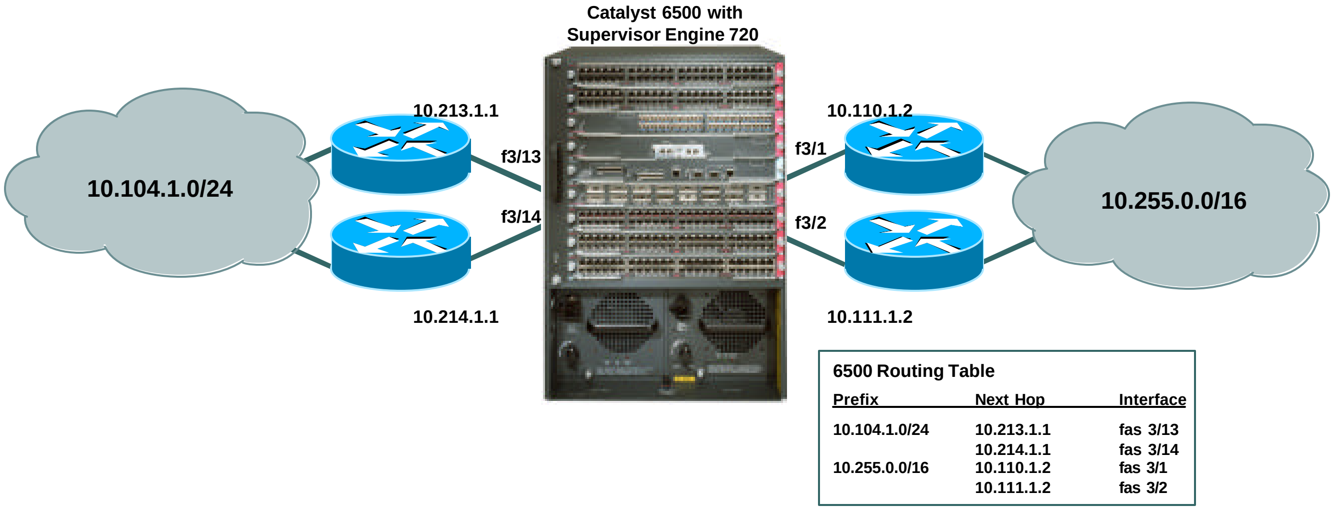
In Scenario 1, where all prefixes have less than three valid reverse-path interfaces, the system programs the global uRPF-check hardware adjacency entry for “drop” mode (for details on the function of this special hardware adjacency entry, see the “Global uRPF-Check Hardware Adjacency” section on page 26):

```
6506#show mls cef ip rpf

RPF global mode:        strict
RPF failed action:      drop
RPF mpath mode:         punt

 Index   Interfaces
-------+----------------------------------------
   0
   1
   2
   3
6506#
```

In drop mode, the large majority of packets hitting the global uRPF check adjacency are dropped in hardware, although in the default configuration, some packets are leaked to the MSFC3 CPU for software visibility (using the global uRPF check hardware-to-CPU rate limiter described later).

In addition, every prefix in the hardware FIB table is programmed with one or two reverse-path VLAN IDs that are used for the reverse-path lookups. For example, consider the topology shown in Figure 4.

Figure 4: Punt Mode Example with Two or Fewer Reverse-Path Interfaces Per Prefix



6500 Routing Table

Prefix	Next Hop	Interface
10.104.1.0/24	10.213.1.1	fas 3/13
	10.214.1.1	fas 3/14
10.255.0.0/16	10.110.1.2	fas 3/1
	10.111.1.2	fas 3/2

The Supervisor Engine 720 has the following routing table:

```
6506#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 8 subnets, 2 masks
C       10.4.0.0/16 is directly connected, GigabitEthernet4/5
S       10.50.0.0/16 [1/0] via 10.4.0.2
O       10.104.1.0/24 [110/2] via 10.213.1.1, 00:00:13, FastEthernet3/13
                      [110/2] via 10.214.1.1, 00:00:13, FastEthernet3/14
C       10.111.1.0/24 is directly connected, FastEthernet3/2
C       10.110.1.0/24 is directly connected, FastEthernet3/1
C       10.214.1.0/24 is directly connected, FastEthernet3/14
C       10.213.1.0/24 is directly connected, FastEthernet3/13
O       10.255.0.0/16 [110/2] via 10.110.1.2, 00:00:13, FastEthernet3/1
                      [110/2] via 10.111.1.2, 00:00:13, FastEthernet3/2
6506#
```

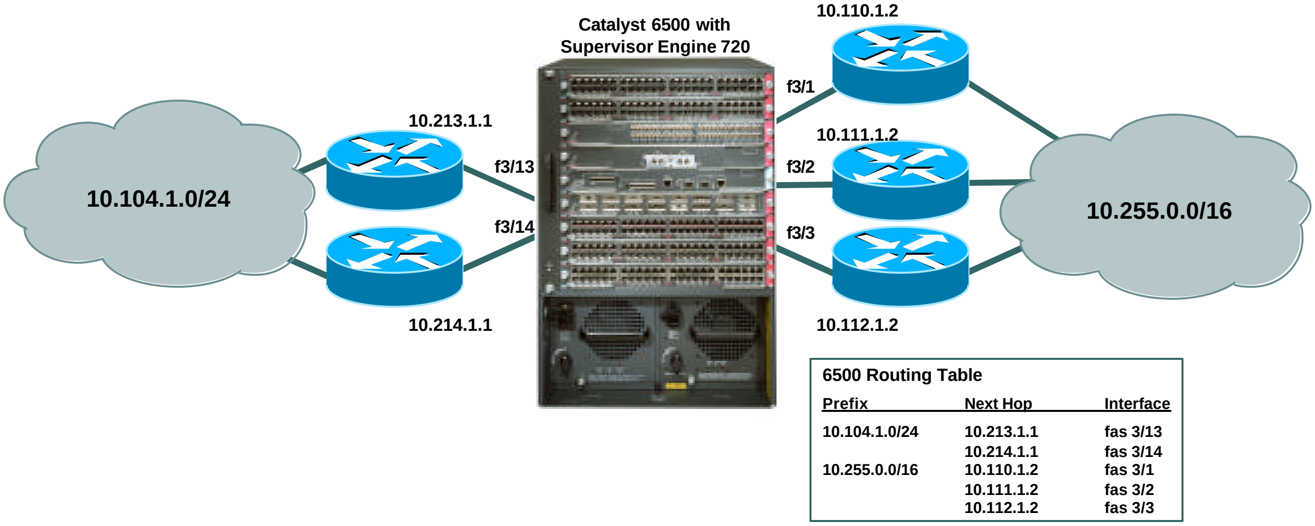
In this example, there are two multipath prefix entries—10.104.1.0/24 and 10.255.0.0/16 (highlighted in red). We can use the **show mls cef ip rpf *prefix*** command to verify which interfaces are programmed as the reverse-path interfaces for these two prefixes:

```
6506#show mls cef ip rpf 10.104.1.0
RPF information for prefix 10.104.1.0/24
uRPF check performed in the hardware for interfaces :
        FastEthernet3/13
        FastEthernet3/14

6506#show mls cef ip rpf 10.255.0.0
RPF information for prefix 10.255.0.0/16
uRPF check performed in the hardware for interfaces :
        FastEthernet3/1
        FastEthernet3/2

6506#
```

Notice the text highlighted in red in the output—for the 10.255.0.0/16 prefix, the two interfaces programmed for hardware uRPF check are FastEthernet 3/1 and FastEthernet 3/2. Therefore, traffic sourced from 10.255.0.0/16 arriving on either of these interfaces will pass the uRPF check in hardware. Traffic sourced from this prefix but arriving on any other interface in the system will be dropped in hardware.

Now we will consider Scenario 2 with punt mode and the strict uRPF check method, where at least one prefix has three or more valid reverse-path interfaces in the routing table. For example, consider the topology shown in Figure 5.

Figure 5: Punt Mode Example with Three or More Reverse-Path Interfaces



6500 Routing Table

Prefix	Next Hop	Interface
10.104.1.0/24	10.213.1.1	fas 3/13
	10.214.1.1	fas 3/14
10.255.0.0/16	10.110.1.2	fas 3/1
	10.111.1.2	fas 3/2
	10.112.1.2	fas 3/3

The Supervisor Engine 720 has the following routing table entry for the 10.255.0.0/16 prefix:

```
6506#show ip route 10.255.0.0
Routing entry for 10.255.0.0/16
  Known via "ospf 100", distance 110, metric 2, type intra area
  Last update from 10.112.1.2 on FastEthernet3/3, 00:12:41 ago
  Routing Descriptor Blocks:
  * 10.110.1.2, from 172.20.45.36, 00:12:41 ago, via FastEthernet3/1
      Route metric is 2, traffic share count is 1
    10.111.1.2, from 172.20.45.36, 00:12:41 ago, via FastEthernet3/2
      Route metric is 2, traffic share count is 1
    10.112.1.2, from 172.20.45.36, 00:12:41 ago, via FastEthernet3/3
      Route metric is 2, traffic share count is 1

6506#
```

Here, the 10.255.0.0/16 prefix has three reverse-path interfaces, FastEthernet 3/1, FastEthernet 3/2, and FastEthernet 3/3. In this case, the system programs the global uRPF-check hardware adjacency entry for “punt” mode:

```
6506#show mls cef ip rpf

RPF global mode:        strict
RPF failed action:      punt
RPF mpath mode:         punt

 Index   Interfaces
-------+---------------------------------------
   0
   1
   2
   3
6506#
```

In punt mode, the system flags all packets hitting the global uRPF check adjacency to be redirected to the MSFC3 CPU. But, as discussed in the "Global uRPF-Check Hardware Adjacency" section on page 26, both uRPF failed packets and packets requiring uRPF check in software will hit this adjacency. In addition, in the default configuration, only a fraction of these packets will actually reach the CPU due to the global uRPF check hardware-to-CPU rate limiter.

In other words, in the default configuration, using uRPF check punt mode when there are prefixes that have more than two valid reverse-path interfaces can cause undesirable behavior—a large number of packets sourced from such prefixes, and arriving on a valid reverse-path interface that is programmed by the system for software uRPF, will be dropped by the hardware-to-CPU rate limiter, thereby affecting throughput of potentially valid traffic.

On the other hand, if you change the rate limiter configuration to allow more packets to hit the MSFC3 CPU, the CPU utilization will rise in proportion to the number of packets that must be processed. Also, some packets being redirected will be true uRPF failed packets, which will then have to be dropped in software.

All prefixes with three or more reverse-path interfaces are programmed with only two reverse-path VLAN IDs that are used for the reverse-path lookups—this is the same as when the hardware adjacency is programmed for drop mode. For example, the 10.255.0.0/16 prefix still shows that FastEthernet 3/1 and FastEthernet 3/2 are programmed for uRPF check in hardware:

```
6506#show mls cef ip rpf 10.255.0.0
RPF information for prefix 10.255.0.0/16
uRPF check performed in the hardware for interfaces :
        FastEthernet3/1
        FastEthernet3/2

6506#
```

Packets arriving on either of these interfaces sourced from the 10.255.0.0/16 prefix will pass the uRPF check in hardware. Packets that fail the uRPF check on either of these interfaces will hit the global uRPF-check hardware adjacency and be redirected to the MSFC3 CPU. In addition, all packets arriving on FastEthernet 3/3, a valid reverse-path interface for the prefix, but one that is *not* programmed for hardware uRPF check, will *also* hit the global adjacency and be redirected to software.

Punt mode and strict uRPF check with allow default

When using the strict uRPF check method with allow default in punt mode, if a default route is present in the routing table, packets sourced from an IP prefix that does not exist in the routing table pass the uRPF check if they are received on a reverse-path interface for the default route. In all other respects, the system behaves as described in the section "Punt mode and strict uRPF check" on page 5. In the same manner as the strict uRPF check method without allow default, if there are prefixes (including the default route) that have more than two reverse-path interfaces, some traffic will hit the uRPF check hardware adjacency and therefore may be dropped by the hardware-to-CPU rate limiter or impact the MSFC3 CPU.

If no default route is present in the routing table, the system behaves exactly as a system using the strict uRPF check method.

Assuming a default route is present in the routing table, a received packet sourced from a prefix that does not exist in the routing table will match the source lookup entry for the default route in the hardware-forwarding table provided the packet arrived on a valid reverse-path interface for the default route:

```
6506#show mls cef ip rpf

RPF global mode:        strict with allow default
RPF failed action:      drop
RPF mpath mode:         punt

 Index   Interfaces
-------+----------------------------------------
   0
   1
   2
   3
6506#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 10.213.1.1 to network 0.0.0.0

     10.0.0.0/8 is variably subnetted, 8 subnets, 2 masks
C       10.4.0.0/16 is directly connected, GigabitEthernet4/5
S       10.50.0.0/16 [1/0] via 10.4.0.2
O       10.104.1.0/24 [110/2] via 10.213.1.1, 00:01:12, FastEthernet3/13
                      [110/2] via 10.214.1.1, 00:01:12, FastEthernet3/14
C       10.111.1.0/24 is directly connected, FastEthernet3/2
C       10.110.1.0/24 is directly connected, FastEthernet3/1
C       10.214.1.0/24 is directly connected, FastEthernet3/14
C       10.213.1.0/24 is directly connected, FastEthernet3/13
O       10.255.0.0/16 [110/2] via 10.111.1.2, 00:01:13, FastEthernet3/2
                      [110/2] via 10.110.1.2, 00:01:13, FastEthernet3/1
O*E2 0.0.0.0/0 [110/1] via 10.213.1.1, 00:00:47, FastEthernet3/13
               [110/1] via 10.214.1.1, 00:00:47, FastEthernet3/14
6506#
```

In this case, the reverse-path interfaces for the default prefix are FastEthernet 3/13 and FastEthernet 3/14. These interfaces are programmed as the reverse-path interfaces for the default route in the hardware. Therefore, any packet that fails to match a more specific entry in the forwarding table will match this entry and pass the uRPF check as long as it arrives on either of these interfaces. If such a packet arrives on any other interface, it fails the uRPF check.

In the event that the default route has three or more valid reverse-path interfaces, the hardware forwarding entry for the default route is programmed in the same manner as above. However, because the “RPF failed action” is now “punt,” packets arriving on the reverse-path interface that is *not* programmed in hardware will hit the global uRPF-check hardware adjacency and be redirected to the CPU:

```
6506#show mls cef ip rpf

RPF global mode:        strict with allow default
RPF failed action:      punt
RPF mpath mode:         punt

 Index   Interfaces
-------+----------------------------------------
```

```
   0
   1
   2
   3
6506#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 10.213.1.1 to network 0.0.0.0

     10.0.0.0/8 is variably subnetted, 9 subnets, 2 masks
C       10.4.0.0/16 is directly connected, GigabitEthernet4/5
S       10.50.0.0/16 [1/0] via 10.4.0.2
O       10.104.1.0/24 [110/2] via 10.213.1.1, 00:25:59, FastEthernet3/13
                      [110/2] via 10.214.1.1, 00:25:59, FastEthernet3/14
                      [110/2] via 10.215.1.1, 00:25:59, FastEthernet3/15
C       10.111.1.0/24 is directly connected, FastEthernet3/2
C       10.110.1.0/24 is directly connected, FastEthernet3/1
C       10.215.1.0/24 is directly connected, FastEthernet3/15
C       10.214.1.0/24 is directly connected, FastEthernet3/14
C       10.213.1.0/24 is directly connected, FastEthernet3/13
O       10.255.0.0/16 [110/2] via 10.111.1.2, 00:25:59, FastEthernet3/2
                      [110/2] via 10.110.1.2, 00:25:59, FastEthernet3/1
O*E2 0.0.0.0/0 [110/1] via 10.213.1.1, 00:25:59, FastEthernet3/13
               [110/1] via 10.214.1.1, 00:25:59, FastEthernet3/14
               [110/1] via 10.215.1.1, 00:26:00, FastEthernet3/15
6506#
```

In this case, only FastEthernet 3/13 and FastEthernet 3/14 are programmed in hardware for the default prefix. Therefore, all packets arriving on FastEthernet 3/15 will be redirected to the CPU for uRPF check processing in software. Remember also that packets hitting the redirect adjacency are rate limited to the MSFC3 CPU and some packets that would have passed the uRPF check in software will likely be dropped in hardware instead.

Punt mode and loose uRPF check

When using the loose uRPF check method in punt mode, the Supervisor Engine 720 supports up to eight reverse-path interfaces per prefix in hardware. The only requirement for a packet received by the system is that a prefix exists in the routing table for the source IP address contained in the packet. Therefore, the uRPF check hardware adjacency is programmed for drop mode:

```
6506#show mls cef ip rpf

RPF global mode:        loose
RPF failed action:      drop
RPF mpath mode:         punt

 Index   Interfaces
```

```
-------+---------------------------------------
   0
   1
   2
   3
6506#
```

In addition, all prefixes in the forwarding table are programmed to not compare the RPF VLAN IDs:

```
6506#show ip route 10.255.0.0
Routing entry for 10.255.0.0/16
  Known via "ospf 100", distance 110, metric 2, type intra area
  Last update from 10.111.1.2 on FastEthernet3/2, 00:16:36 ago
  Routing Descriptor Blocks:
  * 10.111.1.2, from 172.20.45.36, 00:16:36 ago, via FastEthernet3/2
      Route metric is 2, traffic share count is 1
    10.110.1.2, from 172.20.45.36, 00:16:36 ago, via FastEthernet3/1
      Route metric is 2, traffic share count is 1
    10.112.1.2, from 172.20.45.36, 00:16:36 ago, via FastEthernet3/3
      Route metric is 2, traffic share count is 1
6506#
```

Note that even thought the system is in punt mode and the 10.255.0.0/16 prefix has more than two reverse-path interfaces, the uRPF check hardware adjacency is still in drop mode. This is because in loose mode, prefixes are programmed to not compare the source interface of the packet with the reverse-path interfaces associated with the prefix when performing a source IP address lookup.

Punt mode and loose uRPF check with allow default

When using the loose uRPF check method with allow default in punt mode, if there is a default route in the routing table, all packets will pass the uRPF check in hardware. In other words, the system behaves as if uRPF check is completely disabled. However, if no default route exists in the routing table, the system behaves the same as when using the loose uRPF check method.

In the loose uRPF check method with allow default, the default route entry in the forwarding table is configured to ignore the source interface VLAN IDs when performing a source lookup, and the uRPF check hardware adjacency is configured in drop mode:

```
6506#show mls cef ip rpf

RPF global mode:        loose with allow default
RPF failed action:      drop
RPF mpath mode:         punt

 Index   Interfaces
-------+---------------------------------------
   0
   1
   2
   3
6506#
```

Using punt mode—Recommendations

While punt mode with the strict uRPF check method ensures that any packet that is forwarded by the system is subjected to a uRPF check, thus providing the greatest level of uRPF protection, in cases where there are prefixes with more than three reverse paths, punt mode will result in one or more of these problems:

- Some packets that should pass uRPF check will not reach the MSFC3 CPU and will be dropped in hardware by the hardware-to-CPU rate limiter
- Some packets that failed the uRPF check will reach the MSFC3 CPU and be dropped in software
- If the hardware-to-CPU rate limiter is disabled or configured to allow a greater number of packets to reach the MSFC3 CPU, CPU utilization will increase in proportion to the number of packets in either category that are redirected to the CPU

Therefore, **we do not recommend punt mode with the strict uRPF check method for networks that have prefixes with more than two reverse-path interfaces in the routing table**. One simple way to ensure that no more than two reverse-path interfaces exist in the routing table for each prefix is to use the **maximum-paths 2** command in config-router mode when configuring OSPF, EIGRP, or BGP.

If you are using the loose uRPF check method, then there is no downside to using the default punt mode, regardless of whether there are prefixes in the routing table that have three or more reverse-path interfaces. Up to eight reverse-path interfaces are supported.

Pass Mode

In pass mode, Supervisor Engine 720 supports any number of RPF interfaces for every prefix in the routing table in hardware. However, when used with the strict uRPF check method, this mode essentially disables uRPF check for multipath prefixes that have more than two valid reverse-path interfaces.

Pass mode and strict uRPF check method

With the strict uRPF check method in a pass mode configuration, the source lookup for each prefix is programmed as follows:

- If the prefix has less than three valid reverse-path interfaces in the routing table, the source lookup is programmed with one or two reverse-path interfaces used to verify the reverse path
- If the prefix has three or more valid reverse-path interfaces in the routing table, the source lookup is programmed to bypass the uRPF check completely

In pass mode, the system always programs the global uRPF-check hardware adjacency entry for "drop" mode (for details, see the "Global uRPF-Check Hardware Adjacency" section on page 26):

```
6506#show mls cef ip rpf

RPF global mode:        strict
RPF failed action:      drop
RPF mpath mode:         pass

 Index   Interfaces
-------+----------------------------------------
   0
   1
   2
   3
```

```
6506#
```

In drop mode, the large majority of packets hitting the global uRPF check adjacency are dropped in hardware, although in the default configuration, some packets are leaked to the MSFC3 CPU for software visibility (using the global uRPF check hardware-to-CPU rate limiter).

Consider a system with the following routing table:

```
6506#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 9 subnets, 2 masks
C       10.4.0.0/16 is directly connected, GigabitEthernet4/5
S       10.50.0.0/16 [1/0] via 10.4.0.2
O       10.104.1.0/24 [110/2] via 10.214.1.1, 00:00:07, FastEthernet3/14
                      [110/2] via 10.213.1.1, 00:00:07, FastEthernet3/13
C       10.111.1.0/24 is directly connected, FastEthernet3/2
C       10.110.1.0/24 is directly connected, FastEthernet3/1
C       10.112.1.0/24 is directly connected, FastEthernet3/3
C       10.214.1.0/24 is directly connected, FastEthernet3/14
C       10.213.1.0/24 is directly connected, FastEthernet3/13
O       10.255.0.0/16 [110/2] via 10.110.1.2, 00:00:07, FastEthernet3/1
                      [110/2] via 10.111.1.2, 00:00:07, FastEthernet3/2
                      [110/2] via 10.112.1.2, 00:00:07, FastEthernet3/3
6506#
```

Notice the two multipath prefix entries—10.104.1.0/24, which has two reverse-path interfaces, and 10.255.0.0/16, which has three.

As mentioned earlier, prefixes that have only one or two reverse-path interfaces are programmed with one or two reverse-path interfaces used to perform the uRPF check on arriving packets based on the result of the source lookup. The 10.104.1.0/24 prefix is an example of a prefix with two reverse-path interfaces. We can verify the programming for this prefix:

```
6506#show mls cef ip rpf 10.104.1.0
RPF information for prefix 10.104.1.0/24
uRPF check performed in the hardware for interfaces :
        FastEthernet3/13
        FastEthernet3/14

6506#
```

We can see that FastEthernet 3/13 and FastEthernet 3/14 are programmed as the reverse-path interfaces for this prefix. Packets sourced from this prefix arriving on either of these interfaces will pass the uRPF check; if they arrive on any other interface, they will fail the uRPF check and be dropped in hardware by the global uRPF-check hardware adjacency.

In the event that a prefix can be reached through more than two different interfaces, the uRPF check for that prefix is disabled—all traffic sourced from that prefix will pass the uRPF check regardless of the interface on which it was received. For example, the 10.255.0.0/16 prefix is programmed to not compare the input interface for packets sourced from this prefix. All packets sourced from this prefix will pass the uRPF check regardless of the interface on which they arrived.

Pass mode and strict uRPF check with allow default method

When using the strict uRPF check method with allow default in pass mode, if there is a default route in the routing table, packets sourced from an IP prefix not in the routing table pass the uRPF check if they are received on a reverse-path interface for the default route. In all other respects, the system behaves as described in the section “Pass mode and strict uRPF check method” on page 13.

If no default route is present in the routing table, the system behaves exactly as a system using the strict uRPF check method.

Assuming a default route is present in the routing table, a received packet sourced from a prefix that does not exist in the routing table will match the source lookup entry for the default route in the hardware-forwarding table provided the packet arrived on a valid reverse-path interface for the default route:

```
6506#show mls cef ip rpf

RPF global mode:        strict with allow default
RPF failed action:      drop
RPF mpath mode:         pass

 Index   Interfaces
-------+----------------------------------------
   0
   1
   2
   3
6506#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 10.213.1.1 to network 0.0.0.0

     10.0.0.0/8 is variably subnetted, 8 subnets, 2 masks
C       10.4.0.0/16 is directly connected, GigabitEthernet4/5
S       10.50.0.0/16 [1/0] via 10.4.0.2
O       10.104.1.0/24 [110/2] via 10.213.1.1, 00:09:32, FastEthernet3/13
                      [110/2] via 10.214.1.1, 00:09:32, FastEthernet3/14
C       10.111.1.0/24 is directly connected, FastEthernet3/2
C       10.110.1.0/24 is directly connected, FastEthernet3/1
C       10.214.1.0/24 is directly connected, FastEthernet3/14
C       10.213.1.0/24 is directly connected, FastEthernet3/13
O       10.255.0.0/16 [110/2] via 10.111.1.2, 00:09:33, FastEthernet3/2
                      [110/2] via 10.110.1.2, 00:09:33, FastEthernet3/1
O*E2 0.0.0.0/0 [110/1] via 10.213.1.1, 00:09:07, FastEthernet3/13
               [110/1] via 10.214.1.1, 00:09:07, FastEthernet3/14
6506#
```

The default route prefix is programmed with the reverse-path interfaces for the prefix. Therefore, any packet that fails to match a more specific entry in the forwarding table will match this entry and pass the uRPF check as long as it arrives on FastEthernet 3/13 or FastEthernet 3/14. If such a packet arrives on any other interface, it fails the uRPF check.

In the event that the default route is reachable via three or more reverse-path interfaces, the uRPF check is disabled for the default route prefix entry. In this case, *all* packets sourced from an unknown prefix will pass the uRPF check:

```
6506#show mls cef ip rpf

RPF global mode:        strict with allow default
RPF failed action:      drop
RPF mpath mode:         pass

 Index   Interfaces
-------+----------------------------------------
   0
   1
   2
   3
6506#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 10.213.1.1 to network 0.0.0.0

     10.0.0.0/8 is variably subnetted, 9 subnets, 2 masks
C       10.4.0.0/16 is directly connected, GigabitEthernet4/5
S       10.50.0.0/16 [1/0] via 10.4.0.2
O       10.104.1.0/24 [110/2] via 10.213.1.1, 00:35:17, FastEthernet3/13
                      [110/2] via 10.214.1.1, 00:35:17, FastEthernet3/14
                      [110/2] via 10.215.1.1, 00:35:17, FastEthernet3/15
C       10.111.1.0/24 is directly connected, FastEthernet3/2
C       10.110.1.0/24 is directly connected, FastEthernet3/1
C       10.215.1.0/24 is directly connected, FastEthernet3/15
C       10.214.1.0/24 is directly connected, FastEthernet3/14
C       10.213.1.0/24 is directly connected, FastEthernet3/13
O       10.255.0.0/16 [110/2] via 10.111.1.2, 00:35:17, FastEthernet3/2
                      [110/2] via 10.110.1.2, 00:35:17, FastEthernet3/1
O*E2 0.0.0.0/0 [110/1] via 10.213.1.1, 00:35:18, FastEthernet3/13
               [110/1] via 10.214.1.1, 00:35:18, FastEthernet3/14
               [110/1] via 10.215.1.1, 00:35:18, FastEthernet3/15
6506#
```

In this case, the default route prefix is programmed to not compare the source interface of packets sourced from this prefix. Therefore, any packet that does not match a more specific prefix entry will hit this entry and pass the uRPF check.

Pass mode and loose uRPF check method

When using the loose uRPF check method in pass mode, the Supervisor Engine 720 behaves exactly as a system using loose uRPF check in punt mode. As long as a prefix for the source IP address exists, the packet passes the uRPF check in hardware. The uRPF check hardware adjacency is programmed in drop mode, and the source lookup entries in the hardware forwarding table are programmed to ignore the source interface fields:

```
6506#show mls cef ip rpf

RPF global mode:        loose with allow default
RPF failed action:      drop
RPF mpath mode:         pass

 Index   Interfaces
-------+----------------------------------------
   0
   1
   2
   3
6506#show ip route 10.255.0.0
Routing entry for 10.255.0.0/16
  Known via "ospf 100", distance 110, metric 2, type intra area
  Last update from 10.111.1.2 on FastEthernet3/2, 00:49:18 ago
  Routing Descriptor Blocks:
  * 10.111.1.2, from 172.20.45.36, 00:49:18 ago, via FastEthernet3/2
      Route metric is 2, traffic share count is 1
    10.110.1.2, from 172.20.45.36, 00:49:18 ago, via FastEthernet3/1
      Route metric is 2, traffic share count is 1
    10.112.1.2, from 172.20.45.36, 00:49:18 ago, via FastEthernet3/3
      Route metric is 2, traffic share count is 1

6506#
```

Pass mode and loose uRPF check with allow default method

When using the loose uRPF check method with allow default in pass mode, if there is a default route in the routing table, all packets will pass the uRPF check in hardware. In other words, the system behaves as if uRPF check is completely disabled. However, if no default route exists in the routing table, the system behaves the same as when using the loose uRPF check method in pass mode.

With the loose uRPF check method with allow default in pass mode, the default route entry in the forwarding table is configured to ignore the source interface VLAN IDs when performing a source lookup:

```
6506#show mls cef ip rpf

RPF global mode:        loose with allow default
RPF failed action:      drop
RPF mpath mode:         pass

 Index   Interfaces
-------+----------------------------------------
   0
   1
   2
   3
6506#
```

Using pass mode—Recommendations

There are pros and cons to using pass mode with the strict uRPF check method:

- Pass mode ensures that every packet that is subjected to uRPF check and fails is dropped in hardware
- For prefixes with more than three reverse paths, packets sourced from those prefixes will always pass the uRPF check regardless of input interface

In other words, for prefixes with more than three reverse paths, the system essentially performs loose (or “exist only”) uRPF check, even when the global uRPF method is strict. This behavior may or may not be acceptable, depending on the goals of the network engineer. Additionally, if the allow default option is configured with strict mode, and the default route has three or more valid reverse-paths, any packet from an unknown source IP address will pass the uRPF check.

In general, **we do not recommend pass mode for networks using a strict-method uRPF check configuration when there are prefixes with more than two reverse-path interfaces in the routing table**. One simple way to ensure that no more than two reverse-path interfaces exist for each prefix in the routing table is to use the **maximum-paths 2** command in config-router mode when configuring OSPF, EIGRP, or BGP.

If you feel that the global strict method, with loose-method checking for prefixes that have more than three reverse-path interfaces, is acceptable, then pass mode provides a simple way to implement multipath uRPF check when some prefixes have more than two reverse-path interfaces.

If you are using the loose uRPF check method, then there is no downside to using pass mode, regardless of whether there are prefixes in the routing table that have three or more reverse-path interfaces. Up to eight reverse-path interfaces per prefix are supported.

Interface-Group Mode

In interface-group mode, Supervisor Engine 720 can support up to six RPF interfaces for every prefix in the routing table in hardware. However, the user must configure multipath-uRPF interface groups that define on which interfaces multipath prefixes with more than two reverse-path interfaces will receive a uRPF check in hardware. Prefixes that have reverse-path interfaces that do not fall into a configured interface group are programmed in the same way as they are in pass mode—when used with the strict uRPF check method, uRPF check is essentially disabled for these prefixes. Similarly, if no interface groups are configured, the system behaves in the same manner as with pass mode.

Interface-group mode and strict uRPF check method

With the strict uRPF check method in an interface-group mode configuration, the source lookup for each prefix is programmed as follows:

- If the prefix has less than three valid reverse-path interfaces in the routing table, the source lookup is programmed with one or two reverse-path interfaces used to verify uRPF
- If the prefix has three to six valid reverse-path interfaces in the routing table, and at least one of those interfaces is part of a configured multipath interface group, the source lookup is programmed with two reverse-path interfaces used to verify uRPF, and is also flagged to perform an additional lookup in the uRPF check interface-group table for the additional one to four reverse-path interfaces
- If the prefix has three or more valid reverse-path interfaces in the routing table, and no interfaces are part of a configured multipath interface group, the source lookup entry is programmed to bypass the uRPF check completely
- If the prefix has more than six valid reverse-path interfaces in the routing table, and at least one of those interfaces is part of a configured multipath interface group, the source lookup entry is programmed to bypass the uRPF check completely

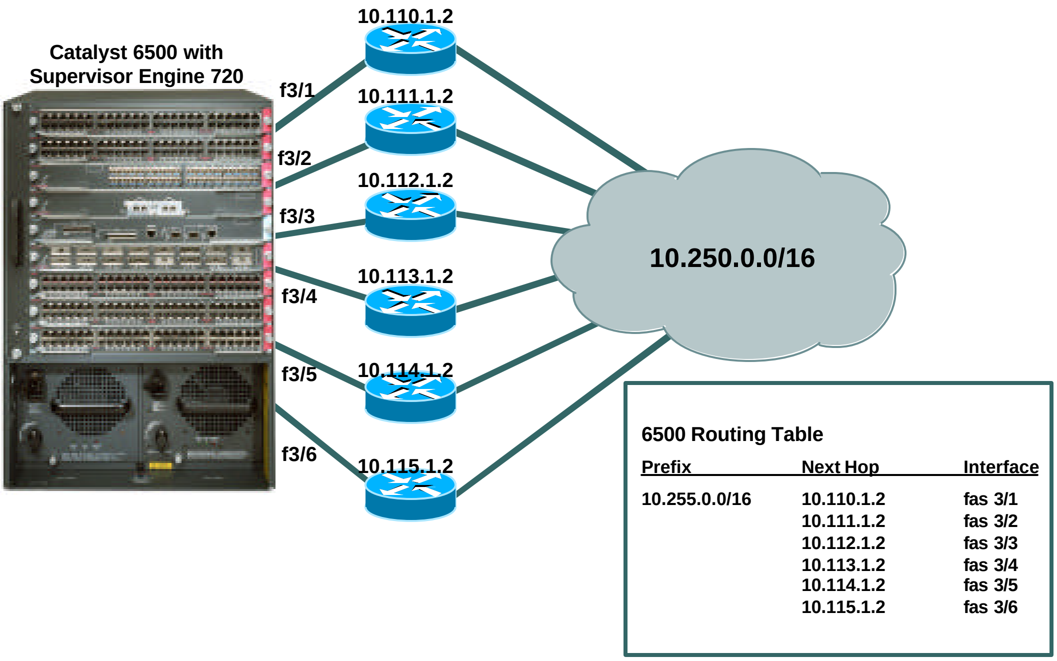
In interface-group mode, the system always programs the global uRPF-check hardware adjacency entry for “drop” mode (for details, see the “Global uRPF-Check Hardware Adjacency” section on page 26):

```
6506#show mls cef ip rpf

RPF global mode:        strict
RPF failed action:      drop
RPF mpath mode:         ifgrp

 Index   Interfaces
-------+---------------------------------------
   0
   1
   2     Fa3/3, Fa3/4, Fa3/5, Fa3/6
   3
6506#
```

Here we see that the system is running in interface-group mode and that one uRPF check interface group (group 2) contains four interfaces, FastEthernet 3/3, Fas tEthernet 3/4, FastEthernet 3/5, and FastEthernet 3/6.

Consider a system running in interface-group mode with the configuration shown above in the topology shown in Figure 6.

Figure 6: Interface-Group Mode Example with Six Reverse-Path Interfaces



6500 Routing Table

Prefix	Next Hop	Interface
10.255.0.0/16	10.110.1.2	fas 3/1
	10.111.1.2	fas 3/2
	10.112.1.2	fas 3/3
	10.113.1.2	fas 3/4
	10.114.1.2	fas 3/5
	10.115.1.2	fas 3/6

The Supervisor Engine 720 has the following routing table:

```
6506#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 10.214.1.1 to network 0.0.0.0

     172.20.0.0/28 is subnetted, 1 subnets
C       172.20.45.32 is directly connected, FastEthernet3/48
     10.0.0.0/8 is variably subnetted, 12 subnets, 2 masks
C       10.2.0.0/16 is directly connected, GigabitEthernet1/14
C       10.4.0.0/16 is directly connected, GigabitEthernet4/5
O       10.104.1.0/24 [110/2] via 10.214.1.1, 00:00:02, FastEthernet3/14
                      [110/2] via 10.213.1.1, 00:00:02, FastEthernet3/13
C       10.111.1.0/24 is directly connected, FastEthernet3/2
C       10.110.1.0/24 is directly connected, FastEthernet3/1
C       10.115.1.0/24 is directly connected, FastEthernet3/6
C       10.114.1.0/24 is directly connected, FastEthernet3/5
C       10.113.1.0/24 is directly connected, FastEthernet3/4
C       10.112.1.0/24 is directly connected, FastEthernet3/3
C       10.214.1.0/24 is directly connected, FastEthernet3/14
C       10.213.1.0/24 is directly connected, FastEthernet3/13
O       10.255.0.0/16 [110/2] via 10.110.1.2, 00:00:03, FastEthernet3/1
                      [110/2] via 10.111.1.2, 00:00:03, FastEthernet3/2
                      [110/2] via 10.112.1.2, 00:00:03, FastEthernet3/3
                      [110/2] via 10.113.1.2, 00:00:03, FastEthernet3/4
                      [110/2] via 10.114.1.2, 00:00:03, FastEthernet3/5
                      [110/2] via 10.115.1.2, 00:00:04, FastEthernet3/6
O*E2 0.0.0.0/0 [110/1] via 10.214.1.1, 00:00:04, FastEthernet3/14
               [110/1] via 10.213.1.1, 00:00:04, FastEthernet3/13
6506#
```

The 10.255.0.0/16 prefix has six valid reverse-path interfaces. Four of these interfaces correspond to the interfaces configured in uRPF check interface group 2. We can verify the programming of the prefix:

```
6506#show mls cef ip rpf 10.255.0.0
RPF information for prefix 10.255.0.0/16
uRPF check performed in the hardware for interfaces :
        FastEthernet3/1
        FastEthernet3/2
uRPF check performed in the hardware for interface-group 2 :
        FastEthernet3/3
        FastEthernet3/4
        FastEthernet3/5
        FastEthernet3/6

6506#
```

Here we see that the first two reverse-path interfaces, FastEthernet 3/1 and FastEthernet 3/2, are programmed for the source lookup. In addition, unlike in punt or pass mode, we can see that this prefix is flagged to perform an additional source interface check in the global uRPF check interface-group table, where the user-configured interface groups are stored. In this case, interface group 2 is selected for this additional lookup.

Therefore, these additional four interfaces, FastEthernet 3/3, FastEthernet 3/4, FastEthernet 3/5, and FastEthernet 3/6, are also compared to the input interface of each packet to verify the reverse path.

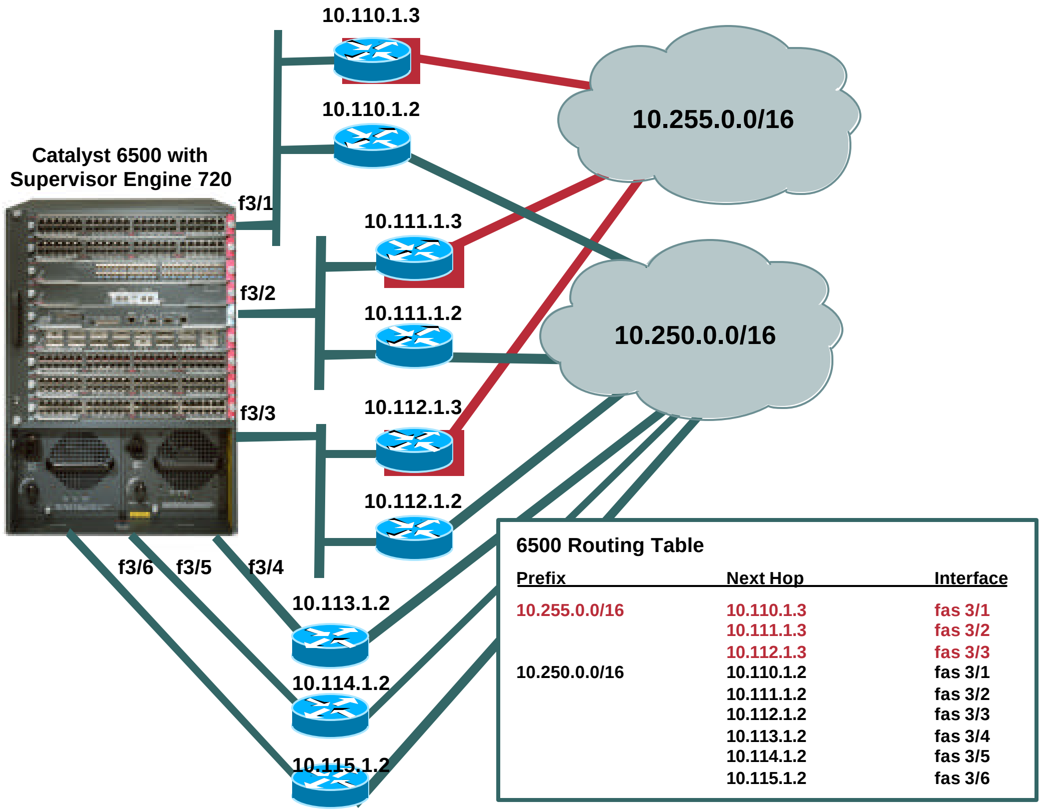
There are two important side effects of using the interface-group mode:

- When a prefix has between one and three valid reverse-path interfaces that are a subset of a configured interface-group (meaning some, but not all, of the interfaces in the interface group are valid reverse paths), packets sourced from that prefix will pass the uRPF check on **any** of the interfaces in the interface group
- When a prefix has more than two additional valid reverse-path interfaces that are not part of a configured interface group, the prefix is configured for pass mode, meaning that uRPF check is disabled for the prefix and all packets sourced from that prefix will pass the uRPF check regardless of the input interface

As an example of the first point, take a router with the following uRPF check configuration in the topology shown in Figure 7.

```
6506#show mls cef ip rpf

RPF global mode:        strict
RPF failed action:      drop
RPF mpath mode:         ifgrp

 Index   Interfaces
-------+----------------------------------------
   0
   1
   2     Fa3/3, Fa3/4, Fa3/5, Fa3/6
   3
6506#
```

Figure 7: Interface-Group Mode Example with Interface-Group Sharing



6500 Routing Table

Prefix	Next Hop	Interface
10.255.0.0/16	10.110.1.3	fas 3/1
	10.111.1.3	fas 3/2
	10.112.1.3	fas 3/3
10.250.0.0/16	10.110.1.2	fas 3/1
	10.111.1.2	fas 3/2
	10.112.1.2	fas 3/3
	10.113.1.2	fas 3/4
	10.114.1.2	fas 3/5
	10.115.1.2	fas 3/6

The Supervisor Engine 720 has the following routing table:

```
6506#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 10.214.1.1 to network 0.0.0.0

     172.20.0.0/28 is subnetted, 1 subnets
C       172.20.45.32 is directly connected, FastEthernet3/48
     10.0.0.0/8 is variably subnetted, 14 subnets, 2 masks
C       10.2.0.0/16 is directly connected, GigabitEthernet1/14
C       10.4.0.0/16 is directly connected, GigabitEthernet4/5
S       10.50.0.0/16 [1/0] via 10.4.0.2
O       10.104.1.0/24 [110/2] via 10.214.1.1, 00:02:28, FastEthernet3/14
                      [110/2] via 10.213.1.1, 00:02:28, FastEthernet3/13
C       10.111.1.0/24 is directly connected, FastEthernet3/2
```

```
C       10.110.1.0/24 is directly connected, FastEthernet3/1
C       10.115.1.0/24 is directly connected, FastEthernet3/6
C       10.114.1.0/24 is directly connected, FastEthernet3/5
C       10.113.1.0/24 is directly connected, FastEthernet3/4
C       10.112.1.0/24 is directly connected, FastEthernet3/3
C       10.214.1.0/24 is directly connected, FastEthernet3/14
C       10.213.1.0/24 is directly connected, FastEthernet3/13
O       10.250.0.0/16 [110/2] via 10.110.1.2, 00:02:29, FastEthernet3/1
                      [110/2] via 10.111.1.2, 00:02:29, FastEthernet3/2
                      [110/2] via 10.112.1.2, 00:02:29, FastEthernet3/3
                      [110/2] via 10.113.1.2, 00:02:29, FastEthernet3/4
                      [110/2] via 10.114.1.2, 00:02:29, FastEthernet3/5
                      [110/2] via 10.115.1.2, 00:02:29, FastEthernet3/6
O IA    10.255.0.0/16 [110/1] via 10.110.1.3, 00:02:29, FastEthernet3/1
                      [110/1] via 10.111.1.3, 00:02:29, FastEthernet3/2
                      [110/1] via 10.112.1.3, 00:02:29, FastEthernet3/3
O*E2 0.0.0.0/0 [110/1] via 10.214.1.1, 00:02:29, FastEthernet3/14
               [110/1] via 10.213.1.1, 00:02:29, FastEthernet3/13
6506#
```

In this case, the 10.250.0.0/16 prefix is reachable through interfaces FastEthernet 3/1 through FastEthernet 3/6. The 10.255.0.0/16 prefix is reachable through interfaces FastEthernet 3/1 through FastEthernet 3/3. With this configuration, packets sourced from prefix 10.255.0.0/16 arriving on interfaces FastEthernet 3/4 through FastEthernet 3/6 will always pass the uRPF check even though those are not valid reverse-path interfaces for the prefix. This is because the 10.255.0.0/16 prefix is programmed to use interface-group number 2, which contains FastEthernet 3/3, FastEthernet 3/4, FastEthernet 3/5, and FastEthernet 3/6. Any prefix that shares one or more interfaces with a configured interface group (and has less than three additional reverse-path interfaces that are not part of the interface group) is programmed to use the entire interface group for verifying the reverse path.

To illustrate, we can look at the programming of the source lookup for the 10.255.0.0/16 prefix:

```
6506#show mls cef ip rpf 10.255.0.0
RPF information for prefix 10.255.0.0/16
uRPF check performed in the hardware for interfaces :
        FastEthernet3/1
        FastEthernet3/2
uRPF check performed in the hardware for interface-group 2 :
        FastEthernet3/3
        FastEthernet3/4
        FastEthernet3/5
        FastEthernet3/6

6506#
```

Notice that this prefix is programmed to use group 2 from the uRPF check interface-group table. In addition, notice that FastEthernet 3/1 and FastEthernet 3/2 are programmed in the source lookup for this prefix. Therefore, traffic sourced from this prefix will pass the uRPF check as long as it arrives on any interface in the range from FastEthernet 3/1 to FastEthernet 3/6, even though the only valid reverse-path interfaces in the routing table are FastEthernet 3/1, FastEthernet 3/2, and FastEthernet 3/3.

Now suppose that the routing table changes such that the 10.255.0.0/16 prefix is reachable through FastEthernet 3/7 as well:

```
6506#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
```

```
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 10.214.1.1 to network 0.0.0.0

     172.20.0.0/28 is subnetted, 1 subnets
C       172.20.45.32 is directly connected, FastEthernet3/48
     10.0.0.0/8 is variably subnetted, 15 subnets, 2 masks
C       10.2.0.0/16 is directly connected, GigabitEthernet1/14
C       10.4.0.0/16 is directly connected, GigabitEthernet4/5
S       10.50.0.0/16 [1/0] via 10.4.0.2
O       10.104.1.0/24 [110/2] via 10.214.1.1, 00:02:14, FastEthernet3/14
                      [110/2] via 10.213.1.1, 00:02:14, FastEthernet3/13
C       10.111.1.0/24 is directly connected, FastEthernet3/2
C       10.110.1.0/24 is directly connected, FastEthernet3/1
C       10.115.1.0/24 is directly connected, FastEthernet3/6
C       10.114.1.0/24 is directly connected, FastEthernet3/5
C       10.113.1.0/24 is directly connected, FastEthernet3/4
C       10.112.1.0/24 is directly connected, FastEthernet3/3
C       10.116.1.0/24 is directly connected, FastEthernet3/7
C       10.214.1.0/24 is directly connected, FastEthernet3/14
C       10.213.1.0/24 is directly connected, FastEthernet3/13
O       10.250.0.0/16 [110/2] via 10.110.1.2, 00:02:15, FastEthernet3/1
                      [110/2] via 10.112.1.2, 00:02:15, FastEthernet3/3
                      [110/2] via 10.113.1.2, 00:02:15, FastEthernet3/4
                      [110/2] via 10.114.1.2, 00:02:15, FastEthernet3/5
                      [110/2] via 10.115.1.2, 00:02:15, FastEthernet3/6
                      [110/2] via 10.111.1.2, 00:02:15, FastEthernet3/2
                      [110/2] via 10.116.1.2, 00:02:16, FastEthernet3/7
O IA    10.255.0.0/16 [110/1] via 10.110.1.3, 00:02:16, FastEthernet3/1
                      [110/1] via 10.111.1.3, 00:02:16, FastEthernet3/2
                      [110/1] via 10.112.1.3, 00:02:16, FastEthernet3/3
                      [110/1] via 10.116.1.3, 00:02:16, FastEthernet3/7
O*E2 0.0.0.0/0 [110/1] via 10.214.1.1, 00:02:16, FastEthernet3/14
               [110/1] via 10.213.1.1, 00:02:16, FastEthernet3/13
S    171.0.0.0/8 [1/0] via 172.20.45.33
6506#
```

Now, the reverse-path interface list for the prefix has three reverse-path interfaces that are not part of the interface group. Therefore, the prefix is programmed in pass mode—that is, uRPF check is disabled for the prefix, and packets sourced from the prefix will always pass the uRPF check.

Interface-group mode and strict uRPF check with allow default method

When using the strict uRPF check method with allow default in interface-group mode, if there is a default route in the routing table, packets sourced from an IP prefix that is not in the routing table will pass the uRPF check if they are received on one of the reverse-path interfaces for the default route. In all other respects, the system behaves as described in the section "Interface-group mode and strict uRPF check method" on page 18.

If no default route is present in the routing table, the system behaves exactly as a system using the strict uRPF check method.

Assuming a default route is present in the routing table, a received packet sourced from a prefix that does not exist in the routing table will match the source lookup entry for the default route in the hardware forwarding table provided the packet arrived on a valid reverse-path interface for the default route. The rules described in the section "Interface-group mode and strict uRPF check method" on page 18 for normal IP prefixes also apply for the default route:

- If the default route has less than three valid reverse-path interfaces in the routing table, the source lookup is programmed with one or two reverse-path interfaces used to verify uRPF
- If the default route has three to six valid reverse-path interfaces in the routing table, and at least one of those interfaces is part of a configured multipath interface group, the source lookup is programmed with two reverse-path interfaces used to verify uRPF, and is also flagged to perform an additional lookup in the uRPF check interface-group table for the additional one to four reverse-path interfaces
- If the default route has three or more valid reverse-path interfaces in the routing table, and no interfaces are part of a configured multipath interface group, the source lookup is programmed to bypass the uRPF check completely
- If the default route has more than six valid reverse-path interfaces in the routing table, and at least one of those interfaces is part of a configured multipath interface group, the source lookup is programmed to bypass the uRPF check completely
- If the default route has between one and three valid reverse-path interfaces that are a subset of a configured interface-group (meaning not all of the interfaces in the interface group are valid reverse paths), packets sourced from unknown prefixes will pass the uRPF check on any of the interfaces in the interface group
- When the default route has more than two additional valid reverse-path interfaces that are not part of a configured interface group, the default route is configured for pass mode, meaning that uRPF check is disabled for the default route and all packets sourced from that unknown prefixes will pass the uRPF check regardless of the input interface

Interface-group mode and loose uRPF check method

When using the loose uRPF check method in interface-group mode, the Supervisor Engine 720 behaves exactly as a system using loose uRPF check in punt or pass mode. As long as a prefix for the source IP address exists, the packet passes the uRPF check in hardware. The uRPF check hardware adjacency is programmed in drop mode, and the source lookup entries in the hardware-forwarding table are programmed to ignore the source interface fields. Therefore, any uRPF check interface groups configured are simply ignored. Interface groups are not used in loose uRPF check method.

Interface-group mode and loose uRPF check with allow default method

When using the loose uRPF check method with allow default in interface-group mode, if there is a default route in the routing table, all packets will pass the uRPF check in hardware because the default route entry in the forwarding table is configured to ignore the source interface VLAN IDs when performing a source lookup. In other words, the system behaves as if uRPF check is completely disabled. However, if no default route exists in the routing table, the system behaves the same as when using the loose uRPF check method in interface-group mode. In any case, any uRPF check interface groups configured are simply ignored. Interface groups are not used in loose uRPF check method.

Using interface-group mode—Recommendations

There are pros and cons to using the interface-group mode:

- Allows uRPF check on up to six reverse-path interfaces per prefix in hardware, and ensures that every packet that is subjected to uRPF check and fails is dropped in hardware
- Correct interface-group configuration and proper implementation are required for interface-group mode to work as desired
- There are operational caveats, as described in preceding sections, that must be fully understood before implementing interface-group mode

In general, **we recommend interface-group mode for networks using a strict-method uRPF check configuration when there are prefixes with more than two and less than six reverse-path interfaces**. One simple way to ensure that no more than six reverse-path interfaces exist for each prefix in the routing table is to use the **maximum-paths 6** command in config-router mode when configuring OSPF, EIGRP, or BGP. However, there are additional requirements for implementing interface-group mode, such as identifying the interfaces you want to configure in the uRPF check interface-group table. This requires an understanding of the routing table as well as all of the possible interfaces involved in routing in your network topology.

If you are using the loose-method uRPF check method, then there is no downside to using interface-group mode, regardless of whether there are prefixes in the routing table that have six or more reverse-path interfaces. Interface-group mode behaves the same as pass mode when using the loose uRPF check method.

Per-Prefix Source IP FIB Table Entries

When you enable uRPF check on the Supervisor Engine 720, a source IP lookup entry is enabled for every IP prefix in the routing table. The system uses these source lookup entries to verify that each input packet arrived on a valid reverse-path interface for the source IP address contained in the packet.

Depending on the global uRPF check mode of the system, these source entries are programmed differently. The different methods and modes are described elsewhere in this document. However, regardless of the programming, the hardware architecture of the system is such that performing both a source IP and a destination IP lookup for each packet does not affect system performance in virtually every hardware configuration—the forwarding engine is designed to function at top speed regardless of whether source IP address lookups are enabled or not.

Global uRPF-Check Hardware Adjacency

There is a single hardware adjacency entry used for unicast RPF check in Supervisor Engine 720. This adjacency serves two purposes, depending on how it is programmed:

- Drop—Drops uRPF check failed packets in hardware
- Punt—Redirects uRPF check failed packets, and packets requiring software uRPF check, to the MSFC3 CPU

How the adjacency is programmed depends on two factors: the global uRPF check mode, and whether there are prefixes with three or more valid reverse-path interfaces in the routing table.

Regardless of how the uRPF check adjacency is programmed in hardware, by default, some packets are "leaked" to the MSFC3 CPU for processing. The number of packets leaked is dependent on the settings configured for the global uRPF hardware-to-CPU rate limiter, which is described in the "Global uRPF Hardware-to-CPU Rate Limiter" section on page 28.

The uRPF check adjacency is programmed in drop mode in the following cases:

- The uRPF check mode is punt and there are no prefixes with three or more reverse-path interfaces

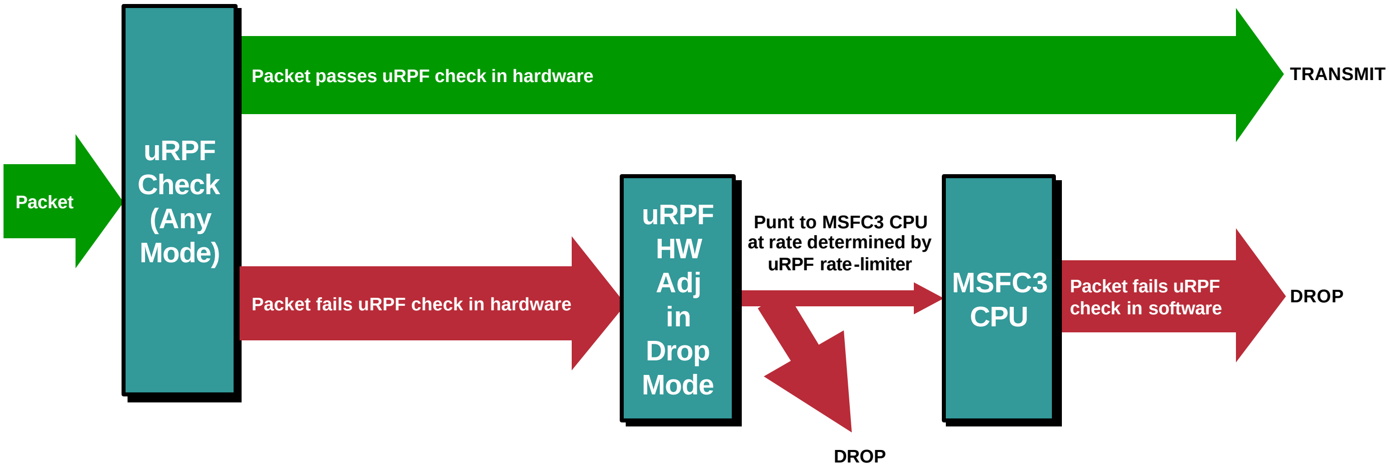
- The uRPF check mode is pass or interface-group

The uRPF check hardware adjacency can only be programmed in drop mode when:

- The system can ensure that hardware uRPF check is possible on every unicast RPF check-enabled interface for every prefix
- The user has chosen a mode (such as pass mode) in which uRPF check is not required for certain prefixes

In drop mode, all packets hitting the adjacency are guaranteed to be uRPF check failed packets. Figure 8 shows the behavior of the uRPF check hardware adjacency when the hardware adjacency is operating in drop mode.

Figure 8: uRPF Check Hardware Adjacency Operating in Drop Mode

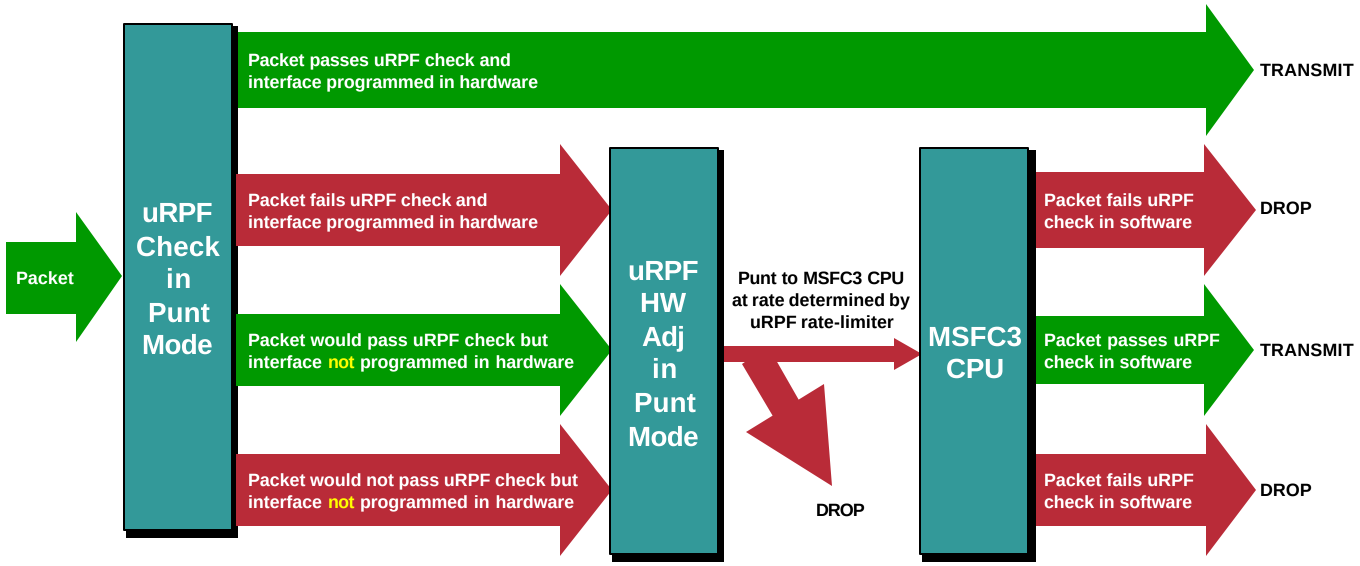

Packet
uRPF Check (Any Mode)
Packet passes uRPF check in hardware
TRANSMIT
Packet fails uRPF check in hardware
uRPF HW Adj in Drop Mode
Punt to MSFC3 CPU at rate determined by uRPF rate-limiter
DROP
MSFC3 CPU
Packet fails uRPF check in software
DROP

The uRPF check adjacency is programmed in punt mode in the following case:

- The uRPF check mode is punt and there is at least one prefix with three or more reverse-path interfaces

The uRPF check hardware adjacency must be programmed in punt mode whenever the system cannot ensure that hardware uRPF check is possible on every unicast RPF check-enabled interface for every prefix. In other words, punt mode is used when some packets entering the system must be verified in the software. In punt mode, some packets hitting the adjacency are uRPF check failed packets, while others might be packets that require uRPF check in software. Figure 9 shows the behavior of the uRPF check hardware adjacency when the hardware adjacency is operating in punt mode because at least one prefix has three or more reverse-path interfaces.

Figure 9: uRPF Check Hardware Adjacency Operating in Punt Mode


Packet
uRPF Check in Punt Mode
Packet passes uRPF check and interface programmed in hardware
TRANSMIT
Packet fails uRPF check and interface programmed in hardware
Packet would pass uRPF check but interface not programmed in hardware
Packet would not pass uRPF check but interface not programmed in hardware
uRPF HW Adj in Punt Mode
Punt to MSFC3 CPU at rate determined by uRPF rate-limiter
DROP
MSFC3 CPU
Packet fails uRPF check in software
DROP
Packet passes uRPF check in software
TRANSMIT
Packet fails uRPF check in software
DROP

Global uRPF Hardware-to-CPU Rate Limiter

Supervisor Engine 720 incorporates several hardware-based rate-limiters that protect the MSFC3 CPU from excessive traffic and DoS attacks. One of these rate limiters is used to leak a certain number of packets that are redirected to the MSFC3 CPU when they hit the global uRPF-check hardware adjacency. This adjacency is described in more detail in the "Global uRPF-Check Hardware Adjacency" section on page 26.

Recall that packets hitting this adjacency might all be uRPF check failed packets (when the adjacency is programmed in drop mode) or they might be a mix of uRPF check failed packets and packets that require uRPF check in software (when the adjacency is programmed in punt mode).

By default, the uRPF hardware-to-CPU rate limiter is enabled with a rate of 500 packets and a burst of 10 packets:

```
6506#show mls rate-limit | include Packet|RPF F
 Rate Limiter Type   Status        Packets/s   Burst
       RPF FAILURE   On                  500      10
6506#
```

In most cases, it is wise to leave the rate limiter at its default configuration so that the MSFC3 CPU is not bothered with excessive uRPF failed packets, especially when the uRPF check hardware adjacency is in drop mode. If the adjacency is in punt mode, you need to consider which is better—leaving the rate limiter at its default but potentially dropping some packets in hardware that would have passed the uRPF check, or disabling the rate limiter but potentially having high CPU utilization.

If you change the rate limiter configuration, pick a rate appropriate to your configuration. For example, if the uRPF check hardware adjacency is in drop mode, very few packets, if any, need to be leaked to the MSFC3 CPU:

```
6506#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
6506(config)#mls rate-limit unicast rpf-failure 1000 10
6506(config)#^Z
```

```
6506#show mls rate-limit | include Packet|RPF F
 Rate Limiter Type   Status       Packets/s   Burst
       RPF FAILURE   On                1000      10
6506#
```

On the other hand, if the uRPF check hardware adjacency is in punt mode, you may want to allow a significantly higher traffic rate to hit the MSFC3 CPU to try to process as many packets as possible (at the expense of high CPU utilization):

```
6506#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
6506(config)#mls rate-limit unicast rpf-failure 50000 255
6506(config)#^Z
6506#show mls rate-limit | include Packet|RPF F
 Rate Limiter Type   Status       Packets/s   Burst
       RPF FAILURE   On               50000     255
6506#
```

A simple performance test of the MSFC3 with the hardware-to-CPU rate limiter completely disabled shows that the CPU can forward approximately 235,000 packets per second in software of valid reverse-path traffic with uRPF check enabled. The test was performed using 64-byte packets and the MSFC3 CPU utilization was at 100%/100%.

One important point to note is that the uRPF check hardware-to-CPU rate limiter is “shared” with two other packet types: unroutable packets and packets requiring ICMP unreachables (ACL deny packets). Therefore, the packet per second and burst values are applied in aggregate on uRPF packets, unroutable packets, and packets denied by an ACL. This example shows a system in which the rate-limiter for all three packet types is enabled:

```
6506#show mls rate-limit usage
                        Rate Limiter Type   Packets/s   Burst
                        -----------------   ---------   -----
Rate Limiter # 0: Free                  -           -       -
Rate Limiter # 1: Free                  -           -       -
Rate Limiter # 2: Free                  -           -       -
Rate Limiter # 3: Used
                           MCAST_DFLT_ADJ      100000     100
Rate Limiter # 4: Free                  -           -       -
Rate Limiter # 5: Free                  -           -       -
Rate Limiter # 6: Used
                              RPF FAILURE       50000     255
                                 NO ROUTE       50000     255
                         ICMP UNREACHABLE       50000     255
Rate Limiter # 7: Used
                                 VACL LOG        2000       1
6506#
```

The use of the uRPF rate limiter does protect the MSFC3 CPU, but because not all packets hitting the uRPF check hardware adjacency reach the CPU, keep the following in mind:

- uRPF statistics are not accurate in the MSFC3 software—Commands such as **show ip traffic** and **show ip interface** will show uRPF check fail counters incrementing, but these numbers do not reflect the total number of packets actually dropped
- When the uRPF check adjacency is in punt mode, packets will be rate limited regardless of whether they would pass the uRPF check in software—Some packets that would pass the uRPF check in software will be dropped before reaching the CPU for processing

Global uRPF Check Interface-Group Table

When you configure the multipath uRPF mode as interface-group, you must specify one or more multipath interface groups for which the system will perform uRPF check in hardware. The uRPF check interface-group table contains one to four interface groups that identify a group of interfaces for which hardware uRPF check should be performed. For details on how the uRPF check interface-group mode functions, see the "Interface-Group Mode" section on page 18.

To configure a uRPF check interface group, use the **mls ip cef rpf interface-group** command:

```
6506#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
6506(config)#mls ip cef rpf interface-group 2 fas 3/3 fas 3/4 fas 3/5 fas 3/6
6506(config)#mls ip cef rpf interface-group 3 fas 3/13 fas 3/14 fas 3/15 fas 3/16
6506(config)#^Z
6506#show mls cef ip rpf

RPF global mode:        loose
RPF failed action:      drop
RPF mpath mode:         ifgrp

 Index   Interfaces
-------+---------------------------------------
   0
   1
   2     Fa3/3, Fa3/4, Fa3/5, Fa3/6
   3     Fa3/13, Fa3/14, Fa3/15, Fa3/16
6506#
```

You can configure up to four different interface groups per system.

uRPF Check CLI and Statistics

There are several uRPF check configuration commands in the Cisco IOS software for Supervisor Engine 720. These configuration commands are identified and summarized in Table 1.

Table 1: uRPF Check Configuration Commands on Supervisor Engine 720

Command	Purpose
ip verify unicast source reachable-via {any|rx} [allow-default] [allow-self-ping]	Interface configuration mode. Enables the different uRPF check methods on an interface. Note that on Supervisor Engine 720, the uRPF check method is **global**, and changes to the last uRPF check method applied to an interface.
mls ip cef rpf multipath {punt|pass|interface-group}	Global configuration mode. Specifies the global uRPF check mode for the system. The default is **punt** mode.
mls ip cef rpf interface-group ***group interface-list***	Global configuration mode. Defines uRPF check multipath interface groups. The *group* parameter is a value from 0-3, while the *interface-*

	list is a list of one to four interfaces.
mls rate-limit unicast rpf-failure *packets burst*	Global configuration mode. Defines the leaking values for the uRPF check hardware-to-CPU rate limiter. The default is 500 packets per second with a 10-packet burst.

Details of the function and behavior of these various commands can be found elsewhere in this document.

Once you have uRPF check configured, there are some additional commands available for verifying the configuration and monitoring the status of uRPF check in the system.

The verification and monitoring commands are identified and summarized in Table 2.

Table 2: uRPF Check Verification and Monitoring Commands on Supervisor Engine 720

Command	Purpose
show ip interface *interface*	Displays uRPF check configuration for the interface, as well as the number of uRPF check failures on the interface. Note that in Supervisor Engine 720, uRPF check failures dropped in hardware are not reflected in the output of this command.
show ip traffic	Displays the global number of uRPF check failures. Again, hardware drops are not reflected.
show mls cef ip rpf	Displays the global uRPF check hardware configuration, including the global uRPF check method, the uRPF check mode, the RPF fail action, and any configured multipath interface groups.
show mls cef ip rpf *prefix*	Displays uRPF check information for the specified prefix, based on a longest-match lookup in the hardware-forwarding table. Shows which interfaces are programmed for hardware uRPF check lookups for the prefix, and whether there is an associated multipath interface group.

Following are some output examples and additional information about the commands described in Table 2.

The **show ip interface** command displays several lines of output relating to uRPF check, provided uRPF check is enabled on the interface. A pipe to include shows just the relevant lines of output:

```
6506#show ip interface Gig 1/1 | include verif
  IP verify source reachable-via RX, allow default, allow self-ping
   4345 verification drops
   0 suppressed verification drops
6506#
```

It is important to remember that only uRPF check failed packets that are dropped in software are reflected in this output. As discussed in the section “Global uRPF-Check Hardware Adjacency,” on page 26, only a small portion of uRPF check failed packets are seen by the MSFC3 CPU; the remainder are dropped in hardware.

Therefore, the output of this command will most likely not return a true count of the number of packets dropped due to uRPF check failure. In most cases, however, you can use this command to get an idea of whether uRPF check failures are happening on the interface or not. However, note that if you completely disable the uRPF check hardware-to-CPU rate limiter, the counters in the **show ip interface** command output will not increment at all.

To clear the counters, use the **clear counters *interface*** command. This command clears all counters for the specified interface, including the uRPF check failure statistics.

While the **show ip interface** command shows per-interface statistics, the output of the **show ip traffic** command displays global uRPF check failure statistics for packets dropped in software:

```
6506#show ip traffic | include RPF
         0 no route, 234233 unicast RPF, 0 forced drop
6506#
```

Again, packets dropped in hardware are not accounted for in this output. Also note that there is no way to clear the statistics in this command output other than rebooting the system.

The **show mls cef ip rpf** command shows the global uRPF check configuration for the Supervisor Engine 720 system:

```
6506#show mls cef ip rpf

RPF global mode:        strict
RPF failed action:      drop
RPF mpath mode:         ifgrp

 Index   Interfaces
-------+---------------------------------------
   0
   1     Gi1/3, Gi1/4, Gi2/1, Gi2/2
   2
   3
6506#
```

The “RPF global mode” shows the currently configured uRPF check method for the system, as specified by the **ip verify unicast source reachable-via** interface configuration command. Recall that because this is a global mode, the current mode is determined by whichever **ip verify unicast source reachable-via** interface command was last applied.

The “RPF failed action” shows the programming of the global hardware uRPF check adjacency and is either **drop** (uRPF check failed packets are dropped in hardware) or **punt** (uRPF check failed packets, and packets on interfaces that are configured for uRPF check but cannot be processed in hardware, are punted to the MSFC3 CPU).

The “RPF mpath mode” shows the currently configured multipath uRPF check mode, as specified with the **mls ip cef rpf multipath** global configuration command.

Finally, the “Index” and “Interfaces” display the configuration of the four user-configurable multipath uRPF check interface groups, as defined by the **mls ip cef rpf interface-group** global configuration command.

When you specify an IP prefix after the **show mls cef ip rpf** command, the system displays prefix-specific uRPF check information:

```
6506#show mls cef ip rpf 10.250.0.0
RPF information for prefix 10.250.0.0/16
```

```
uRPF check performed in the hardware for interfaces :
        GigabitEthernet1/1
        GigabitEthernet1/2
uRPF check performed in the hardware for interface-group 1 :
        GigabitEthernet1/3
        GigabitEthernet1/4
        GigabitEthernet2/1
        GigabitEthernet2/2

6506#
```

This command displays all interfaces for which uRPF check will occur in hardware for packets sourced from this prefix. In this case, uRPF check will occur in hardware for the two interfaces GigabitEthernet 1/1 and GigabitEthernet 1/2. In addition, multipath interface group 1 is also associated with the prefix, providing hardware uRPF check for four additional interfaces, GigabitEthernet 1/3, GigabitEthernet 1/4, GigabitEthernet 2/1, and GigabitEthernet 2/2.

Conclusion

Supervisor Engine 720 reaches a new level in hardware-based source-IP verification, raising the bar for uRPF check by delivering key features such as:

- Hardware support for up to six reverse-path interfaces per prefix in the routing table in strict mode with all 256K hardware forwarding entries available for destination prefixes
- Zero-configuration option for two-path uRPF check for every prefix
- Powerful flexibility of multiple user-defined uRPF-check interface groups to support up to six reverse-path interfaces per prefix
- Source IP verification for every packet passing through the system with no performance penalty

Table 3 summarizes the uRPF check configuration recommendations discussed earlier in this document.

Table 3: Summary of uRPF Check Configuration Recommendations

Network Topology	Required uRPF Check Method	Recommended uRPF Check Configuration
All prefixes have one or two reverse-path interfaces in the routing table	Strict or strict with allow-default	Punt mode or pass mode
	Loose or loose with allow-default	Punt mode or pass mode
One or more prefixes have three or more reverse-path interfaces in the routing table	Strict or strict with allow-default, and ignoring uRPF check for prefixes with more than three reverse-path interfaces is acceptable	Pass mode
	Strict or strict with allow-default, and imperative that all packets get uRPF check	Interface-group mode with appropriate interface groups configured
	Loose or loose with allow-default	Pass mode

One or more prefixes have seven or more reverse-path interfaces in the routing table	Any	Pass mode or interface-group mode with appropriate interface groups configured

For more detail on these recommendations, refer to the following sections of this document:

- Using punt mode—Recommendations, page 13
- Using pass mode—Recommendations, page 18
- Using interface-group mode—Recommendations, page 25