Implementing ARP

Address resolution is the process of mapping network addresses to Media Access Control (MAC) addresses. This process is accomplished using the Address Resolution Protocol (ARP). This module describes how to configure ARP processes on the Cisco ASR 9000 Series Aggregation Services Router.

Note For a complete description of the ARP commands listed in this module, refer to the *IP Addresses and Services Command Reference for Cisco ASR 9000 Series Routers*.

Feature History for Configuring ARP

Release	Modification
Release 3.7.2	This feature was introduced.

Prerequisites for Configuring ARP

- You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Restrictions for Configuring ARP

The following restrictions apply to configuring ARP :

- Reverse Address Resolution Protocol (RARP) is not supported.

- Due to a hardware limitation in the Ethernet SPA interfaces installed on all routers, when a packet contains a wrong destination address, the corresponding SPA drops the packet even if the ingress packet count is already incremented in the output of the **show interfaces** command.

The following additional restrictions apply when configuring the Direct Attached Gateway Redundancy (DAGR) feature on Cisco ASR 9000 Series Routers:

- IPv6 is not supported.
- Ethernet bundles are not supported.
- Non-Ethernet interfaces are not supported.
- Hitless ARP Process Restart is not supported.
- Hitless RSP Failover is not supported.

Information About Configuring ARP

To configure ARP, you must understand the following concepts:

IP Addressing Overview

A device in the IP can have both a local address (which uniquely identifies the device on its local segment or LAN) and a network address (which identifies the network to which the device belongs). The local address is more properly known as a *data link address*, because it is contained in the data link layer (Layer 2 of the OSI model) part of the packet header and is read by data-link devices (bridges and all device interfaces, for example). The more technically inclined person will refer to local addresses as *MAC addresses*, because the MAC sublayer within the data link layer processes addresses for the layer.

To communicate with a device on Ethernet, for example, Cisco IOS XR software first must determine the 48-bit MAC or local data-link address of that device. The process of determining the local data-link address from an IP address is called *address resolution*.

Address Resolution on a Single LAN

The following process describes address resolution when the source and destination devices are attached to the same LAN:

1. End System A broadcasts an ARP request onto the LAN, attempting to learn the MAC address of End System B.
2. The broadcast is received and processed by all devices on the LAN, including End System B.
3. Only End System B replies to the ARP request. It sends an ARP reply containing its MAC address to End System A.
4. End System A receives the reply and saves the MAC address of End System B in its ARP cache. (The ARP cache is where network addresses are associated with MAC addresses.)
5. Whenever End System A needs to communicate with End System B, it checks the ARP cache, finds the MAC address of System B, and sends the frame directly, without needing to first use an ARP request.

Address Resolution When Interconnected by a Router

The following process describes address resolution when the source and destination devices are attached to different LANs that are interconnected by a router (only if proxy-arp is turned on):

1. End System Y broadcasts an ARP request onto the LAN, attempting to learn the MAC address of End System Z.
2. The broadcast is received and processed by all devices on the LAN, including Router X.
3. Router X checks its routing table and finds that End System Z is located on a different LAN.
4. Router X therefore acts as a proxy for End System Z. It replies to the ARP request from End System Y, sending an ARP reply containing its own MAC address as if it belonged to End System Z.
5. End System Y receives the ARP reply and saves the MAC address of Router X in its ARP cache, in the entry for End System Z.
6. When End System Y needs to communicate with End System Z, it checks the ARP cache, finds the MAC address of Router X, and sends the frame directly, without using ARP requests.
7. Router X receives the traffic from End System Y and forwards it to End System Z on the other LAN.

ARP and Proxy ARP

Two forms of address resolution are supported by Cisco IOS XR software: Address Resolution Protocol (ARP) and proxy ARP, as defined in RFC 826 and RFC 1027, respectively. Cisco IOS XR software also supports a form of ARP called local proxy ARP.

ARP is used to associate IP addresses with media or MAC addresses. Taking an IP address as input, ARP determines the associated media address. After a media or MAC address is determined, the IP address or media address association is stored in an ARP cache for rapid retrieval. Then the IP datagram is encapsulated in a link-layer frame and sent over the network.

When proxy ARP is disabled, the networking device responds to ARP requests received on an interface only if one of the following conditions is met:

- The target IP address in the ARP request is the same as the interface IP address on which the request is received.
- The target IP address in the ARP request has a statically configured ARP alias.

When proxy ARP is enabled, the networking device also responds to ARP requests that meet all the following conditions:

- The target IP address is not on the same physical network (LAN) on which the request is received.
- The networking device has one or more routes to the target IP address.
- All of the routes to the target IP address go through interfaces other than the one on which the request is received.

When local proxy ARP is enabled, the networking device responds to ARP requests that meet all the following conditions:

- The target IP address in the ARP request, the IP address of the ARP source, and the IP address of the interface on which the ARP request is received are on the same Layer 3 network.
- The next hop for the target IP address is through the same interface as the request is received.

Typically, local proxy ARP is used to resolve MAC addresses to IP addresses in the same Layer 3 network such as, private VLANs that are Layer 2-separated. Local proxy ARP supports all types of interfaces supported by ARP and unnumbered interfaces.

ARP Cache Entries

ARP establishes correspondences between network addresses (an IP address, for example) and Ethernet hardware addresses. A record of each correspondence is kept in a cache for a predetermined amount of time and then discarded.

You can also add a static (permanent) entry to the ARP cache that persists until expressly removed.

From Release 6.5.1 onwards, the supported ARP scale has been increased from 128K to 256K entries per LC CPU. This increase in scale improves performance while multiple ARP operations are being processed on the device.

Direct Attached Gateway Redundancy

Direct Attached Gateway Redundancy (DAGR) allows third-party redundancy schemes on connected devices to use gratuitous ARP as a failover signal, enabling the ARP process to advertise an new type of route in the Routing Information Base (RIB). These routes are distributed by Open Shortest Path First (OSPF).

Sometimes part of an IP network requires redundancy without routing protocols. A prime example is in the mobile environment, where devices such as base station controllers and multimedia gateways are deployed in redundant pairs, with aggressive failover requirements (subsecond or less), but typically do not have the capability to use native Layer 3 protocols such as OSPF or Intermediate System-to-Intermediate System (IS-IS) protocol to manage this redundancy. Instead, these devices assume they are connected to adjacent IP devices over an Ethernet switch, and manage their redundancy at Layer 2, using proprietary mechanisms similar to Virtual Router Redundancy Protocol (VRRP). This requires a resilient Ethernet switching capability, and depends on mechanisms such as MAC learning and MAC flooding.

DAGR is a feature that enables many of these devices to connect directly to Cisco ASR 9000 Series Routers without an intervening Ethernet switch. DAGR enables the subsecond failover requirements to be met using a Layer 3 solution. No MAC learning, flooding, or switching is required.

Note Since mobile devices' 1:1 Layer 2 redundancy mechanisms are proprietary, they do not necessarily conform to any standard. So although most IP mobile equipment is compatible with DAGR, interoperability does require qualification, due to the possibly proprietary nature of the Layer 2 mechanisms with which DAGR interfaces.

Additional Guidelines

The following are additional guidelines to consider when configuring DAGR:

- Up to 40 DAGR peers, which may be on the same or different interfaces, are supported per system.

- Failover is supported for DAGR routes within 500 ms of receipt of an ARP reply packet.
- On ARP process restart, DAGR groups are reinitialized.

How to Configure ARP

This section contains instructions for the following tasks:

Defining a Static ARP Cache Entry

ARP and other address resolution protocols provide a dynamic mapping between IP addresses and media addresses. Because most hosts support dynamic address resolution, generally you need not to specify static ARP cache entries. If you must define them, you can do so globally. Performing this task installs a permanent entry in the ARP cache. Cisco IOS XR software uses this entry to translate 32-bit IP addresses into 48-bit hardware addresses.

Note From Release 6.5.1 onwards, the supported ARP scale has been increased from 128K to 256K entries per LC CPU. This increase in scale improves performance while multiple ARP operations are being processed on the device.

Optionally, you can specify that the software responds to ARP requests as if it were the owner of the specified IP address by making an alias entry in the ARP cache.

SUMMARY STEPS

1. **configure**
2. Do one of the following:
   - **arp** [**vrf** *vrf-name*] *ip-address hardware-address encapsulation-type*
   - **arp** [**vrf** *vrf-name*] *ip-address hardware-address encapsulation-type* **alias**
3. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	Do one of the following: • **arp** [**vrf** *vrf-name*] *ip-address hardware-address encapsulation-type* • **arp** [**vrf** *vrf-name*] *ip-address hardware-address encapsulation-type* **alias** **Example:** `RP/0/RSP0/CPU0:router(config)# arp 192.168.7.19 0800.0900.1834 arpa`	Creates a static ARP cache entry associating the specified 32-bit IP address with the specified 48-bit hardware address. **Note** If an **alias** entry is created, then any interface to which the entry is attached will act as if it is the owner of the specified addresses, that is, it will respond to ARP request packets for this network layer address with the data link layer address in the entry.

	Command or Action	Purpose
	or `RP/0/RSP0/CPU0:router(config)# arp 192.168.7.19 0800.0900.1834 arpa alias`	
Step 3	**commit**	

Enabling Proxy ARP

Cisco IOS XR software uses proxy ARP (as defined in RFC 1027) to help hosts with no knowledge of routing determine the media addresses of hosts on other networks or subnets. For example, if the router receives an ARP request for a host that is not on the same interface as the ARP request sender, and if the router has all of its routes to that host through other interfaces, then it generates a proxy ARP reply packet giving its own local data-link address. The host that sent the ARP request then sends its packets to the router, which forwards them to the intended host. Proxy ARP is disabled by default; this task describes how to enable proxy ARP if it has been disabled.

SUMMARY STEPS

1. **configure**
2. **interface** *type number*
3. proxy-arp
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**interface** *type number* **Example:** `RP/0/RSP0/CPU0:router(config)# interface MgmtEth 0/RSP0/CPU0/0`	Enters interface configuration mode.
Step 3	proxy-arp **Example:** `RP/0/RSP0/CPU0:router(config-if)# proxy-arp`	Enables proxy ARP on the interface.
Step 4	**commit**	

Enabling Local Proxy ARP

Local proxy ARP is disabled by default; this task describes how to enable local proxy ARP.

SUMMARY STEPS

1. **configure**
2. **interface** *type number*
3. **local-proxy-arp**
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**interface** *type number* **Example:** `RP/0/RSP0/CPU0:router(config)# interface TenGigE 0/0/0/0`	Enters interface configuration mode.
Step 3	**local-proxy-arp** **Example:** `RP/0/RSP0/CPU0:router(config-if)# local-proxy-arp`	Enables local proxy ARP on the interface.
Step 4	**commit**	

Configuring DAGR

Follow these steps to create a DAGR group on the Cisco ASR 9000 Series Router.

SUMMARY STEPS

1. configure
2. **interface** *type interface-path-id*
3. arp dagr
4. **peer ipv4** *address*
5. **route distance normal** *normal- distance* **priority** *priority-distance*
6. **route metric normal** *normal- metric* **priority** *priority-metric*
7. **timers query** *query-time* **standby** *standby-time*
8. **priority-timeout** *time*
9. Do one of the following:
   - end
   - commit
10. **show arp dagr** [*interface* [*IP-address*]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface** *type interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config)# interface gigabitethernet 0/2/0/0`	Enters interface configuration mode and configures an interface.
Step 3	arp dagr **Example:** `RP/0/RSP0/CPU0:router(config-if)# arp dagr`	Enters DAGR configuration mode.
Step 4	**peer ipv4** *address* **Example:** `RP/0/RSP0/CPU0:router(config-if-dagr)# peer ipv4 10.0.0.100`	Creates a new DAGR group for the virtual IP address.
Step 5	**route distance normal** *normal- distance* **priority** *priority-distance* **Example:** `RP/0/RSP0/CPU0:router(config-if-dagr-peer)# route distance normal 140 priority 3`	(Optional) Configures route distance for the DAGR group.
Step 6	**route metric normal** *normal- metric* **priority** *priority-metric* **Example:** `RP/0/RSP0/CPU0:router(config-if-dagr-peer)# route metric normal 84 priority 80`	(Optional) Configures the route metric for the DAGR group.
Step 7	**timers query** *query-time* **standby** *standby-time* **Example:** `RP/0/RSP0/CPU0:router(config-if-dagr-peer)# timers query 2 standby 19`	(Optional) Configures the time in seconds between successive ARP requests being sent out for the virtual IP address.
Step 8	**priority-timeout** *time* **Example:** `RP/0/RSP0/CPU0:router(config-if-dagr-peer)# priority-timeout 25`	(Optional) Configures a timer for the length of time in seconds to wait before reverting to normal priority from a high-priority DAGR route.

	Command or Action	Purpose

Step 9	Do one of the following:
	• end
	• commit
	Example:
	`RP/0/RSP0/CPU0:router(config-if-dagr)# end`
	or
	`RP/0/RSP0/CPU0:router(config-if-dagr)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)?[cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Step 10	**show arp dagr** [*interface* [*IP-address*]] **Example:** `RP/0/RSP0/CPU0:router# show arp dagr`	(Optional) Displays the operational state of all DAGR groups. Using the optional *interface* and *IP-address* arguments restricts the output to a specific interface or virtual IP address.

Configuring ARP purge-delay

With Equal Cost Multi Path (ECMP), traffic is load balanced across multiple paths with equal cost. This should provide resiliency against interface flaps. If an interface goes down, the traffic is then routed via the other interface without traffic loss. However, if the first interface comes up, traffic is routed back over it but forwarding will only resume once ARP has been (re)resolved and the adjcency (re)installed. Here a short unexpected interface flap causes this traffic loss and is particularly undesirable.

The purge-delay feature allows existing dynamic entries to persist rather than immediately delete entries which could cause traffic loss following an interface flap.

The purge delay feature works by caching existing dynamic ARP entries when an interface goes down and starting a purge delay timer. When the interface is brought back and the purge delay timer not yet fired, the entries are reinstalled as before. The normal entry timeout is reduced in order to re-ARP for the entries after any interface state change related churn has died down; should the purge delay timer fire before the interface comes back up, the entries are deleted from the cache.

SUMMARY STEPS

1. **configure**
2.
3.
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**Example:** `RP/0/RSP0/CPU0:router(config)# interface MgmtEth 0/`	Enters interface configuration mode.
Step 3	**Example:** `RP/0/RSP0/CPU0:router(config-if)# arp purge-delay 100`	Sets the purge delay time interval.
Step 4	**commit**	

Configuring ARP timeout

Dynamic ARP entries which are learnt by ARP address resolution (when valid ARP replies are received) are timed out every 4 hours by default in order to remove stale entries.

ARP entries that correspond to the local interface or that are statically configured by the user never time out.

SUMMARY STEPS

1.
2.
3.
4. Do one of the following:
   - end
   - commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.

	Command or Action	Purpose
Step 2	**Example:** `RP/0/RSP0/CPU0:router(config)# interface MgmtEth 0/`	Enters interface configuration mode.
Step 3	**Example:** `RP/0/RSP0/CPU0:router(config-if)# arp timeout 100`	Sets the ARP cache timeout interval.

Step 4	Do one of the following:
	• end
	• commit
	Example:
	`RP/0/RSP0/CPU0:router(config-if)# end`
	or
	`RP/0/RSP0/CPU0:router(config-if)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)?[cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configure Learning of Local ARP Entries

You can configure an interface or a sub-interface to learn only the ARP entries from its local subnet.

Note From Release 6.5.1 onwards, the supported ARP scale has been increased from 128K to 256K entries per LC CPU. This increase in scale improves performance while multiple ARP operations are being processed on the device.

Use the following procedure to configure local ARP learning on an interface.

1. Enter the interface configuration mode.

   ```
   Router(config)# interface GigabitEthernet 0/0/0/1
   ```

2. Configure the IPv4/IPv6 address for the interface.

```
Router(config-if)# ipv4 address 12.1.3.4 255.255.255.0
```

3. Configure local ARP learning on the interface.

```
Router(config-if)# arp learning local
```

4. Enable the interface and commit your configuration.

```
Router(config-if)# no shut
Router(config-if)# commit
RP/0/0/CPU0:Dec 12 13:41:16.580 : ifmgr[397]: %PKT_INFRA-LINK-3-UPDOWN : interface
GigabitEthernet 0/0/0/1, changed state to Down
RP/0/0/CPU0:Dec 12 13:41:16.683 : ifmgr[397]: %PKT_INFRA-LINK-3-UPDOWN : interface
GigabitEthernet 0/0/0/1 changed state to Up
```

5. Confirm your configuration.

```
Router(config-if)# show running-configuration
..
Building configuration...
!! IOS XR Configuration 0.0.0
!! Last configuration change at Mon Dec 12 13:41:16 2016
!interface GigabitEthernet 0/0/0/1
 ipv4 address 12.1.3.4 255.255.255.0
 arp learning local
!
```

6. Verify if local ARP learning is working as configured on the interface.

```
Router(config-if)# do show arp idb gigabitEthernet 0/1/0/0 location 0/1/CPU0
Mon Nov 26 14:09:38.898 IST

interface GigabitEthernet 0/1/0/0 (0x00804060):
  IDB Client: default
  IPv4 address 1.1.1.1, Vrf ID 0x60000000
  VRF Name default
  Dynamic learning: Enable
  Dynamic entry timeout: 14400 secs
  Drop adjacency timeout: 3600 secs
  Purge delay: off
  IPv4 caps added (state up)
  MPLS caps not added
  Interface not virtual, not client fwd ref,
  Proxy arp not configured, not enabled
  Local Proxy arp not configured
  Packet IO layer is SPIO
  Srg Role : DEFAULT
  Idb Flag : 49292
  IDB is Complete
  IDB Flag Description:
  [CAPS | COMPLETE | IPV4_CAPS_CREATED | SPIO_ATTACHED |
   SPIO_SUPPORTED]
```

7. (Optional) You can monitor the ARP traffic on the interface.

```
Router(config-if)# do show arp traffic gigabitEthernet 0/1/0/0 location 0/1/CPU0
Mon Nov 26 14:08:34.403 IST

ARP statistics:
  Recv: 0 requests, 0 replies (0 unsolicited)
  Sent: 5 requests, 1 replies (0 proxy, 0 local proxy, 1 gratuitous)
  Subscriber Interface:
         0 requests recv, 0 replies sent, 0 gratuitous replies sent
  Resolve requests rcvd: 1
```

```
  Resolve requests dropped: 0
  Errors: 0 out of memory, 0 no buffers, 0 out of sunbet

ARP cache:
  Total ARP entries in cache: 2
  Dynamic: 0, Interface: 1, Standby: 0
  Alias: 0,   Static: 0,    DHCP: 0,    DropAdj: 1

  IP Packet drop count for GigabitEthernet0_1_0_0: 1
```

Configuration Examples for ARP Configuration on Cisco IOS XR Software

This section provides the following ARP configuration examples:

Creating a Static ARP Cache Entry: Example

The following is an example of a static ARP entry for a typical Ethernet host:

```
configure
arp 192.168.7.19 0800.0900.1834 arpa
```

The following is an example of a static ARP entry for a typical Ethernet host where the software responds to ARP requests as if it were the owner of both the specified IP address and hardware address, whether proxy ARP is enabled or not:

```
configure
arp 192.168.7.19 0800.0900.1834 arpa alias
```

The following is an example of configuring a static arp entry on an SRP device:

```
configure
arp 192.168.8.20 0800.0900.1723 srp
```

Enabling Proxy ARP: Example

The following is an example of enabling proxy ARP:

```
configure
interface MgmtEth 0/
RSP0


/CPU0/0
proxy-arp
```

Displaying the ARP Table: Example

The following example shows how to display the ARP table:

```
Router# show arp

------------------------------------------------------------------------------
0/1/CPU0
------------------------------------------------------------------------------
Address         Age        Hardware Addr   State      Type  Interface
1.1.1.1         -          027d.42e9.bd36  Interface  ARPA  GigabitEthernet0/1/0/0
1.1.1.2         00:00:06   0000.0000.0000  DropAdj    ARPA  GigabitEthernet0/1/0/0
```

Enabling DAGR and Configuring a DAGR Group: Example

The following is an example of enabling DAGR and configuring a DAGR group peer:

```
configure
 interface gigabitethernet 0/1/0/0.1
  arp dagr
   peer ipv4 192.168.7.19
    priority-timeout 25
    route distance normal 48 priority 5
    route metric normal 48 priority 5
    timers query 2 standby 40
    commit
```

Displaying the Operational State of DAGR Groups: Example

The following example shows how to display the current operational state of the DAGR groups:

```
RP/0/RSP0/CPU0:router# show arp dagr
------------------------------------------------------------------------------
0/1/CPU0
------------------------------------------------------------------------------
Interface                Virtual IP      State     Query-pd Dist Metr
GigabitEthernet0/1/0/2     10.168.7.19    Active    None     150  100
GigabitEthernet0/1/0/2     10.24.0.45     Query     1        None None
GigabitEtherget0/1/0/3     10.66.0.45     Init      None     None None
```

ARP Throttling

When remote devices scan for destinations that do not exist in the locally connected network, the packets with unresolved ARP requests causes continuous queue of ARP packets pending for resolution. Failed ARP resolution entries impacts forwarding and performance of the router because CPU cycles are consumed when packets are sent for ARP resolution continuously. ARP throttling prevents unresolved packet queuing at the first hop counter for ARP resolution by adding drop adjacencies for such destinations. A router drops packets for which drop adjacency entries are added. Packets for which adjacency entries are added are forwarded to the next hop. Adjacency and drop adjacency entries are added for destinations in the ARP table of every router that forwards traffic.

You can enable ARP throttling for an interface of any router that forwards traffic to the next hop. If ARP resolution fails for any packet on that interface, that packet is added as an entry for drop adjacency in the forwarding plane of the router for a specified period of timeout value. Therefore, until the configured value of timeout gets over, the traffic hitting drop adjacency is dropped at the router where ARP throttling is configured and the packets are not queued up for ARP. Timeout value is configured in seconds. The default timeout value is 1 hour or 3600 seconds. Once timeout is over for the drop adjacency, ARP deletes the drop

adjacency entry from the ARP database of the router. ARP also sends a message to Adjacency Information Base (AIB) to delete it. AIB is a database of adjacencies that are learned from the ARP database and AIB provides information to Forwarding Information Base (FIB) that has a database of adjacencies and static routes. Static routes are configuration based and resides in Routing Information Base (RIB) that is linked to the FIB.

Restrictions

- You can configure ARP throttling only on interfaces and not on nodes.
- The entries for drop adjacencies are not retained in the ARP database if there is an interface flap.

Configuration Example

To configure ARP throttling on an interface with specified timeout for drop adjacency, complete the following configurations:

1. Enter interface configuration mode.
2. Configure ARP throttling on the interface for a specified timeout period for drop adjacency.

Configuration

```
/* Enter the global configuration mode and configure an interface */
Router# config
Router(config)# interface GigabitEthernet 0/1/0/0

/* Configure ARP throttling on the interface with specified timeout for drop adjacency. */
Router(config-if)# arp drop-adjacency timeout 1200
Router(config-if)# commit
```

Verification

To verify the drop adjacencies in the ARP database with respect to interfaces, use the **show arp** command. To verify the drop adjacency traffic statistics in the ARP cache, use the **show arp traffic** command.

To verify the different types of adjacencies on a router's interface, use the **show adjacency summary** command:

```
 Router# show adjacency summary location 0/1/CPU0
Mon Nov 26 14:10:25.352 IST

Adjacency table (version 7) has 7 adjacencies:
      2 complete adjacencies
      0 incomplete adjacencies
      5 interface adjacencies
      0 deleted adjacencies in quarantine list
      2 adjacencies of type IPv4
          2 complete adjacencies of type IPv4
          0 incomplete adjacencies of type IPv4
          1 drop adjacency of type IPv4
          0 deleted adjacencies of type IPv4 in quarantine list
          1 multicast adjacency of type IPv4
```

To verify the details of each type of adjacency, use the **show adjacency** ***interface*** **internal detail** command. In the output, **Entry-flag: 0x1000** shows that drop adjacencies are identified in the configured interface.

```
RP/0/0/CPU0:ios#show adjacency gigabitEthernet 0/1/0/0 internal detail
Mon Nov 26 14:10:57.440 IST
------------------------------------------------------------------------------
```

```
0/1/CPU0
------------------------------------------------------------------------------

GigabitEthernet0/1/0/0, (src mac only) (ipv4)
        Version: 6, references: 2, transient lock: 0
        Encapsulation information (14 bytes) 000000000000027d42e9bd360800
        MTU: 1500
        Adjacency pointer is: 0x571990ac
        Platform adjacency pointer is: 0
        Last updated: Nov 26 14:07:17.695
        Adjacency producer: arp (prod_id: 10)
        Flags: incomplete adj,
                (Base-flag: 0x1, Entry-flag: 0x1)
        Netio idb pointer not cached
        Cached interface type: 15
        Adjacency references:
                ipv4_mfwd_partner (JID 274, PID 44110), 1 reference
                aib (JID 178, PID 44107), 1 reference

Gi0/1/0/0, (interface)
        Version: 1, references: 1, transient lock: 0
        MTU: 1500
        Adjacency pointer is: 0x57198c60
        Platform adjacency pointer is: 0
        Last updated: Nov 26 14:06:57.267
        Adjacency producer: dot1q (prod_id: 11)
        Flags: interface adjacency, incomplete adj,
                (Base-flag: 0x1, Entry-flag: 0x4)
        Netio idb pointer not cached
        Cached interface type: 15
        Adjacency references:
                aib (JID 178, PID 44107), 1 reference

GigabitEthernet0/1/0/0, 1.1.1.2 (ipv4)
        Version: 7, references: 2, transient lock: 0
        Encapsulation information (14 bytes) 000000000000027d42e9bd360800
        MTU: 1500
        Adjacency pointer is: 0x57199264
        Platform adjacency pointer is: 0
        Last updated: Nov 26 14:07:36.813
        Adjacency producer: arp (prod_id: 10)
        Entry-flag: 0x1000)(Base-flag: 0x1, Entry-flag: 0x1000)
        Netio idb pointer not cached
        Cached interface type: 15
        Adjacency references:
                l2fib_mgr (JID 247, PID 44514), 0 reference
                fib_mgr (JID 261, PID 44119), 1 reference
                aib (JID 178, PID 44107), 1 reference
```

To verify the drop adjacencies in FIB, use the **show cef adjacency {*interface*|*location*}** command:

```
Router# show cef adjacency gigabitEthernet 0/1/0/0 location 0/1/CPU0
Mon Nov 26 14:12:49.924 IST
Display protocol is ipv4
Interface    Address                                          Type    Refcount

Gi0/1/0/0                                                     special 2
             Interface: Gi0/1/0/0 Type:  glean
             Interface Type: 0xf, Base Flags: 0x10001100 (0x573819b8)
             Nhinfo PT: 0x573819b8, Idb PT: 0x5716e354, If Handle: 0x804060
             Dependent adj type: remote (0x57f88060)
             Dependent adj intf: Gi0/1/0/0
             Ancestor If Handle: 0x0
Update time Nov 26 14:07:17.717
```

```
Gi0/1/0/0     Prefix: 1.1.1.2/32                              local   3
              Adjacency: PT:0x57199264 1.1.1.2/32
              Interface: Gi0/1/0/0
              NHID: 0x0
              MAC: 00.00.00.00.00.00.02.7d.42.e9.bd.36.08.00
              Interface Type: 0xf, Base Flags: 0x30000001 (0x57f880b8)
              Nhinfo PT: 0x57f880b8, Idb PT: 0x5716e354, If Handle: 0x804060
              Dependent adj type: remote (0x57f88060)
              Dependent adj intf: Gi0/1/0/0
              Ancestor If Handle: 0x0
Update time Nov 26 14:07:36.836
```

To verify the number of drop adjacency packets that are forwarded to the FIB, use the **show cef drops location** ***location-value*** command:

```
Router# ping 1.1.1.2 count 5
Thu Feb 7 19:31:25.893 IST
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
RP/0/RSP0/CPU0:RR#show cef drops  location 0/1/cPU0
Mon Nov 26 09:11:58.669 UTC
CEF Drop Statistics
Node: 0/1/CPU0
  Unresolved drops     packets :               0
  Unsupported drops    packets :               0
  Null0 drops          packets :               0
  No route drops       packets :               0
  No Adjacency drops   packets :               5
  Checksum error drops packets :               0
  RPF drops            packets :               0
  RPF suppressed drops packets :               0
  RP destined drops    packets :               3
  Discard drops        packets :               5
  GRE lookup drops     packets :               0
  GRE processing drops packets :               0
  LISP punt drops      packets :               0
  LISP encap err drops packets :               0
  LISP decap err drops packets :               0

**This counter will get incremented for the traffic received for drop-adjacency only if
interface is configured with "ipv4 unreachabel disable".
             Sample config:
RP/0/RSP0/CPU0:RR#show running-config interface GigabitEthernet 0/1/0/0
Thu Feb 7 19:31:25.893 IST
interface GigabitEthernet0/1/0/0
ipv4 address 1.1.1.1 255.255.255.0
arp drop-adjacency timeout 1200
ipv4 unreachables disable
!
```

To verify the global statistics of the packets that are sent to ICMP instead of ARP, use the **show controllers np counters** ***np-value*** **location** ***location-value*** command. Packets sent to ICMP do not require ARP resolution because they have a drop adjacency ot normal adjacency entry in the FIB.

```
RP/0/RSP0/CPU0:RR#ping 1.1.1.2
Mon Nov 26 09:15:27.370 UTC
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
RP/0/RSP0/CPU0:RR#show controllers np counters np0 location 0/1/cPU0
```

```
Mon Nov 26 09:15:43.651 UTC

                Node: 0/1/CPU0:
----------------------------------------------------------------

Show global stats counters for NP0, revision v3

Last clearing of counters for this NP: NEVER

Read 53 non-zero NP counters:
Offset  Counter                                                 FrameValue   Rate (pps)
-----------------------------------------------------------------------------------------
  16  MDF_TX_LC_CPU                                              3381479           5
  17  MDF_TX_WIRE                                                 963536           0
  21  MDF_TX_FABRIC                                              1824309           0
  33  PARSE_FAB_RECEIVE_CNT                                      1842113           0
  37  PARSE_INTR_RECEIVE_CNT                                   295405833         514
  41  PARSE_INJ_RECEIVE_CNT                                       68023           0
 45  PARSE_ENET_RECEIVE_CNT                                      927472           0
  49  PARSE_TM_LOOP_RECEIVE_CNT                                 11044944          19
  64  DBG_RSV_EP_L_RSV_ING_L3_IFIB                                927244           0
  65  DBG_RSV_EP_L_RSV_ING_L3_IFIB_MATCH                          927244           0
  66  DBG_RSV_EP_L_RSV_ING_L3_IFIB_PUNT_LOCAL                      31654           0
  68  DBG_RSV_EP_L_RSV_ING_L3_RSLTS_MATCH                          927244           0
  69  DBG_RSV_EP_L_RSV_ING_PUNT                                    4287668           4
142  RSV_DROP_IPV4_DROP_RP_DEST                                        16           0
143  RSV_DROP_IPV4_DROP_RP_DEST_MONITOR                                 2           0
149  RSV_DROP_IPV6_RXADJ_DROP_MONITOR                                   1           0
173  RSV_DROP_IPV4_TXADJ_DROP                                          10           0
272  RSV_DROP_IN_L3_NOT_MYMAC                                           5           0
581  MODIFY_PUNT_REASON_MISS_DROP                                       2           0
776  PUNT_ARP                                                          94           0
790  PUNT_DIAGS                                                      9535           0
832  PUNT_FOR_ICMP                                                    104           0
850  PUNT_ADJ                                                        8279           0
862  PUNT_IPV6_HOP_BY_HOP                                              18           0
902  PUNT_STATISTICS                                                3331795           4
904  PUNT_DIAGS_RSP_ACT                                              9463           0
906  PUNT_DIAGS_RSP_STBY                                             9462           0
1084  DROP_FRM_RUNT                                                     2           0
1683  DISCARD_FRM_ERR_INTF_194                                          1           0
1694  DISCARD_CRC_ERR_INTF_198                                          1           0
1695  DISCARD_FRM_ERR_INTF_198                                          1           0
1869  PRS_HEALTH_MON                                                11044932          19
1879  INTR_FRAME_TYPE_7                                              2629849           4
1885  INTR_FRAME_TYPE_13                                              701947           0
1902  PARSE_ING_IPV6_LINK_LOCAL                                        18938           0
1906  PARSE_RSP_INJ_FAB_CNT                                            18008           0
1907  PARSE_RSP_INJ_PORT_CNT                                              51           0
1909  PARSE_RSP_INJ_DIAGS_CNT                                          18925           0
1911  PARSE_EGR_INJ_PKT_TYP_IPV4                                         176           0
1912  PARSE_EGR_INJ_PKT_TYP_IPV6                                          10           0
1915  PARSE_EGR_INJ_PKT_TYP_IPV4_PREROUTE                              17822           0
1922  PARSE_LC_INJ_FAB_CNT                                              7231           0
1923  PARSE_LC_INJ_PORT_CNT                                            51253           0
1924  PARSE_LC_INJ_DIAGS_CNT                                            9535           0
1926  PARSE_DROP_UNKNOWN_TIMER_FEAT                                        3           0
1929  PARSE_FAB_INJ_IPV4_L3_INWARD                                    451473           0
1930  PARSE_FAB_INJ_IPV6_L3_INWARD                                    451097           0
1945  PARSE_DROP_IN_UIDB_DOWN                                              5           0
1979  PARSE_DROP_IPV6_DISABLED                                            36           0
1981  PARSE_DROP_IPV6_LENGTH                                               4           0
2127  ING_ICFD_QUEUE_PRI_0                                            662786           0
2128  ING_ICFD_QUEUE_PRI_1                                            264566           0
```

```
2130  ING_ICFD_QUEUE_PRI_3                                    120           0

**The above counter will get incremented for traffic received for a drop-adjacency.
```

Clearing ARP Cache of Drop Adjacencies

You can delete the drop adjacencies on an interface, location, or an IP address by using the command **clear arp-cache drop-adjacency *interface ip-address location***. This command deletes the drop adjacencies from the ARP database and AIB. To get more information, see the *clear arp-cache* command in the chapter *ARP Commands* of the *IP Addresses and Services Command Reference for Cisco ASR 9000 Series Routers*.

Restrictions

- Configuration of the **clear arp-cache drop-adjacency** command on a particular location is not recommended. If the command is used on a bundle interface that comprises of a few interfaces on few line cards, then drop adjacencies may be deleted in one of the interfaces line cards and not on other line cards. This scenario can result in entry mismatch. You can use the **clear arp-cache drop-adjacency *interface location* all** command to remove drop adjacency that is learned for the interface on all the line cards.

Installing Drop Adjacencies in Hardware

If ARP throttling is configured on an interface and ARP fails to resolve a dynamic entry on that interface, then ARP marks that entry as drop adjacency entry for the interface in its database and in AIB. The drop adjacency timeout starts for that entry at that interface or hardware as per the configured timeout value. Once the timeout period is over, drop adjacency for that entry is removed from the ARP database and AIB.

Handling Drop Adjacencies Over Virtual Interfaces

If ARP throttling is configured over a virtual interface, drop adjacencies are synced with other interfaces like BVI or Bundle Interfaces over Group Service Process (GSP). This synching occurs in the member line cards in case of Bundles and in all the line cards in case of BVIs. Therefore, when an entry state is changed to drop adjacency or removed from drop adjacency, ARPs running on all the line cards have the same drop adjacency state for a particular entry. The drop adjacency state for an entry is updated in AIB as well.

Handling Drop Adjacencies on Process Restart

ARP stores all adjacencies in the shared memory. Hence, after a process restart or an AIB disconnect followed by an AIB connect, ARP restores all drop adjacencies along with dynamic entries to the AIB.

Note The timers for ARP entries, including timeout for drop adjacencies, is reset after a process restart. Therefore, the duration for drop adjacency entries for timeout is increased.

Handling Drop Adjacencies over ISSU and Geo Redundancy

During ISSU, the drop adjacencies in the ARP database are not synched from the earlier version of the router to the upgraded version of the router. The drop adjacencies are recreated in the refreshed ARP database and AIB of the upgraded router.

Similarly, if ARP throttling and Geo Redundancy are configured for an interface of a router, the drop adjacencies are not synched across the active and backup routers. Once the backup router becomes the active router, the drop adjacencies are revreated in the refreshed ARP database and AIB of the newly active router.

Handling Drop Adjacencies on Interface Flap

In there is an interface flap, the purge delay feature allows the retention of dynamic entries in the ARP database up to a configured timeout period. The dynamic entries are not deleted from the ARP database if the interface comes up within the configured timeout period. However, the entries for drop adjacencies are not retained in the ARP database if there is an interface flap.

Additional References

The following sections provide references related to ARP.

Related Documents

Related Topic	Document Title
ARP commands	*ARP Commands* module in *IP Addresses and Services Command Reference for Cisco ASR 9000 Series Routers*
Getting started material	*Cisco ASR 9000 Series Aggregation Services Router Getting Started Guide*

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Quality of Service Commands* module in *Modular Quality of Service Command Reference for Cisco ASR 9000 Series Routers*
Class-based traffic shaping, traffic policing, low latency queuing, and MDDR	Configuring Modular Quality of Service Congestion Management module in *Modular QoS Configuration Guide for Cisco ASR 9000 Series Routers*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
—	To locate and download MIBs, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: https://mibs.cloudapps.cisco.com/ITDIT/MIBS/servlet/index

RFCs

RFCs	Title
RFC 826	Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware
RFC 1027	Using ARP to implement transparent subnet gateways

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport