Implementing IP in IP Tunnel

This chapter module provides conceptual and configuration information for IP in IP tunnels on Cisco ASR 9000 Series Router.

For a complete description of the IP in IP tunnel commands listed in this chapter, see the Cisco ASR 9000 Series Aggregation Services Router VPN and Ethernet Services Command Reference. To locate documentation of other commands that appear in this chapter, use the command reference master index, or search online.

Table 1: Feature History for IP in IP tunnel

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 5.3.1</td>
<td>This feature was introduced on Cisco ASR 9000 Series Router.</td>
</tr>
</tbody>
</table>

- IP in IP Tunneling, on page 1
- Configuring IP in IP Tunnel, on page 2
- IP in IP Tunneling: Examples, on page 3

IP in IP Tunneling

IP in IP tunneling refers to the encapsulation of an IP packet as a payload in another IP packet. ASR9K routers support IP in IP tunnels with all possible combinations of IPv4 and IPv6; that is, IPv4 over IPv4, IPv6 over IPv4, IPv4 over IPv6, and IPv6 over IPv6. For example, an IPv4 over IPv6 refers to an IPv4 packet as a payload encapsulated within an IPv6 packet and routed across an IPv6 network to reach the destination IPv4 network, where it is decapsulated.

IP in IP tunneling does not require any additional header such as a GRE header used in the GRE tunnels. So, IP in IP tunneling is preferred over GRE tunnels if both the networks are IP networks.

Restrictions

The following are not supported in IP in IP tunnels:

- MPLS
• Multicast packets
• Keep-Alive packets
• Path MTU (Maximum Transmission Unit) discovery
• DF (Do not Fragment) bit configuration in IPv6 tunnel mode.

**Note**
If DF bit is configured for the tunnel interface, you cannot enable IPv6 tunnel mode.

### Configuring IP in IP Tunnel

Perform the following steps to configure an IP in IP tunnel.

**SUMMARY STEPS**

1. `configure`
2. `interface tunnel-ip tunnel-id`
3. `{ipv4 | ipv6} address ip-address`
4. `tunnel mode {ipv4 | ipv6}`
5. `tunnel source {interface-id | ipv4/ipv6-address}`
6. `tunnel destination ipv4/ipv6-address`
7. `(Optional) tunnel df-bit {copy | disable}`
8. `(Optional) tunnel tos tos-value`
9. `commit`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure</td>
<td>Enters the Global Configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>interface tunnel-ip tunnel-id</td>
<td>Creates a tunnel interface and enters the tunnel configuration sub-mode. The <code>tunnel-id</code> is the numeric identifier for the tunnel interface.</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config)# interface tunnel-ip 1</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>`{ipv4</td>
<td>ipv6} address ip-address`</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-if)# ipv6 address 10::1/64</td>
<td></td>
</tr>
</tbody>
</table>
### Purpose

Command or Action: `tunnel mode {ipv4 | ipv6}`

Example: `RP/0/RSP0/CPU0:router(config-if)# tunnel mode ipv6`

- Sets the tunnel mode as IPv4 or IPv6. This states the tunnel is in an IPv4 or IPv6 transport network.

### Step 5

Command or Action: `tunnel source {interface-id | ipv4/v6-address}`

Example: `RP/0/RSP0/CPU0:router(config-if)# tunnel source GigabitEthernet0/0/0/0`

- Specifies an IP address or an interface that serves as the tunnel source. The encapsulated packet uses this IP address as the source address. If the tunnel mode is set to IPv4 or IPv6, an IPv4 or IPv6 address is selected as a tunnel source address from the specified source interface respectively.

### Step 6

Command or Action: `tunnel destination ipv4/v6-address`

Example: `RP/0/RSP0/CPU0:router(config-if)# tunnel destination 100::2`

- Specifies the destination IP address for the tunnel. The encapsulated packet uses this IP address as the destination address.

### Step 7

Command or Action: `(Optional) tunnel df-bit {copy | disable}`

Example: `RP/0/RSP0/CPU0:router(config-if)# tunnel df-bit disable`

- Note: This is valid only for a tunnel that uses an IPv4 transport network. Configures the DF bit value for the outer IP packet. For details on this `tunnel df-bit` command, see the Cisco ASR 9000 Series Aggregation Services Router VPN and Ethernet Services Command Reference.

### Step 8

Command or Action: `(Optional) tunnel tos tos-value`

Example: `RP/0/RSP0/CPU0:router(config-if)# tunnel tos 1`

- Sets the TOS value for the outer IP packet in the tunnel. For details on this `tunnel tos` command, see the Cisco ASR 9000 Series Aggregation Services Router VPN and Ethernet Services Command Reference.

### Step 9

Command or Action: `commit`

---

### IP in IP Tunneling: Examples

The following examples provide configurations for an IPv4 or IPv6 tunnel, with the transport VRF as the default VRF for the following simplified network topology.

*Figure 1: IP in IP tunnel network topology*

![IP in IP tunnel network topology](image)

**Configuration example for an IPv4 tunnel**

---

Implementing IP in IP Tunnel
### PE1 Router Configuration

```c
interface GigabitEthernet0/0/0/0
  !! Link between PE1-PE2
  ipv4 address 100.1.1.1/64
!
interface GigabitEthernet0/0/0/1
  !! Link between CE1-PE1
  ipv4 address 20.1.1.1/24
  ipv6 address 20::1/64
!
interface tunnel-ip 1
  ipv4 address 10.1.1.2/24
  ipv6 address 10::1/64
  tunnel mode ipv4
  tunnel source GigabitEthernet0/0/0/0
  tunnel destination 100.1.1.2
!
router static
  address-family ipv4 unicast
    30.1.1.0/24 tunnel-ip1
  address-family ipv6 unicast
    30::0/64 tunnel-ip1
!
```

### PE2 Router Configuration

```c
interface GigabitEthernet0/0/0/0
  !! Link between PE1-PE2
  ipv4 address 100.1.1.2/64
!
interface GigabitEthernet0/0/0/1
  !! Link between PE2-CE2
  ipv4 address 30.1.1.1/24
  ipv6 address 30::1/64
!
interface tunnel-ip 1
  ipv4 address 10.1.1.2/24
  ipv6 address 10::2/64
  tunnel mode ipv4
  tunnel source GigabitEthernet0/0/0/0
  tunnel destination 100.1.1.1
!
router static
  address-family ipv4 unicast
    20.1.1.0/24 30.1.1.1
  address-family ipv6 unicast
    20::0/64 30::1
!
```

### CE1 Router Configuration

```c
interface GigabitEthernet0/0/0/1
  !! Link between CE1-PE1
  ipv4 address 20.1.1.2 255.255.255.0
  ipv6 address 20::2/64
!
router static
  address-family ipv4 unicast
    30.1.1.0/24 20.1.1.1
  address-family ipv6 unicast
    30::0/64 20::1
!
```

### CE2 Router Configuration

```c
interface GigabitEthernet0/0/0/1
  !! Link between CE2-PE2
  ipv4 address 30.1.1.2 255.255.255.0
  ipv6 address 30::2/64
!
router static
  address-family ipv4 unicast
    20.1.1.0/24 30.1.1.1
  address-family ipv6 unicast
    20::0/64 30::1
!
```

### Configuration example for an IPv6 tunnel

<table>
<thead>
<tr>
<th>PE1 Router Configuration</th>
<th>PE2 Router Configuration</th>
</tr>
</thead>
</table>
| interface GigabitEthernet0/0/0/0
  !! Link between PE1-PE2
  ipv4 address 100.1.1.1/64
  
  interface GigabitEthernet0/0/0/1
  !! Link between PE2-CE2
  ipv4 address 30.1.1.1/24
  ipv6 address 30::1/64
  
  interface tunnel-ip 1
  ipv4 address 10.1.1.2/24
  ipv6 address 10::2/64
  tunnel mode ipv4
  tunnel source GigabitEthernet0/0/0/0
  tunnel destination 100.1.1.1

  router static
  address-family ipv4 unicast
    30.1.1.0/24 tunnel-ip1
  address-family ipv6 unicast
    30::0/64 tunnel-ip1
|
interface GigabitEthernet0/0/0/0
  ! Link between PE1-PE2
  ipv6 address 100::1/64
!
interface GigabitEthernet0/0/0/1
  ! Link between CE1-PE1
  vrf RED
  ipv4 address 20.1.1.1/24
  ipv6 address 20::1/64
!
interface tunnel-ip 1
  vrf RED
  ipv4 address 10.1.1.2/24
  ipv6 address 10::2/64
  tunnel mode ipv6
  tunnel source GigabitEthernet0/0/0/0
  tunnel destination 100::2
!
  vrf RED
  address-family ipv6 unicast
    import route-target 2:1
    !
    export route-target 2:1
    !
  address-family ipv4 unicast
    import route-target 2:1
    !
    export route-target 2:1
    !
  router static
  vrf RED
    address-family ipv4 unicast
      30.1.1.0/24 tunnel-ip1
    address-family ipv6 unicast
      30::0/64 tunnel-ip1
    !
    !

CE1 Router Configuration

interface GigabitEthernet0/0/0/1
  ! Link between CE1-PE1
  ipv4 address 20.1.1.2 255.255.255.0
  ipv6 address 20::2/64
!
  router static
  address-family ipv4 unicast
    30.1.1.0/24 20.1.1.1
  address-family ipv6 unicast
    30::0/64 20::1
!
!

interface GigabitEthernet0/0/0/0/0
  ! Link between PE1-PE2
  ipv6 address 100::2/64
!
interface GigabitEthernet0/0/0/0/1
  ! Link between PE2-CE2
  vrf RED
  ipv4 address 30.1.1.1/24
  ipv6 address 30::1/64
!
interface tunnel-ip 1
  vrf RED
  ipv4 address 10.1.1.2/24
  ipv6 address 10::2/64
  tunnel mode ipv6
  tunnel source GigabitEthernet0/0/0/0
  tunnel destination 100::1
!
  vrf RED
  address-family ipv6 unicast
    import route-target 2:1
    !
    export route-target 2:1
    !
  address-family ipv4 unicast
    import route-target 2:1
    !
    export route-target 2:1
    !
  router static
  vrf RED
    address-family ipv4 unicast
      20.1.1.0/24 tunnel-ip1
    address-family ipv6 unicast
      20::0/64 tunnel-ip1
    !
    !

CE2 Router Configuration

interface GigabitEthernet0/0/0/1
  ! Link between CE2-PE2
  ipv4 address 30.1.1.2 255.255.255.0
  ipv6 address 30::2/64
!
  router static
  address-family ipv4 unicast
    20.1.1.0/24 30.1.1.1
  address-family ipv6 unicast
    20::0/64 30::1
!
!