



Configure IP-in-IP Tunnels

Tunneling provides a mechanism to transport packets of one protocol within another protocol. This chapter describes IP-in-IP tunneling protocol.

Feature History for Configure Tunnels

Release 7.0.11	This feature was introduced.
Release 7.0.14	Support for the following feature was introduced in Configure Tunnels: <ul style="list-style-type: none">• Extended ACL needs to match on the outer header for IP-in-IP De-capsulation.

- [IP-in-IP De-capsulation, on page 1](#)

IP-in-IP De-capsulation

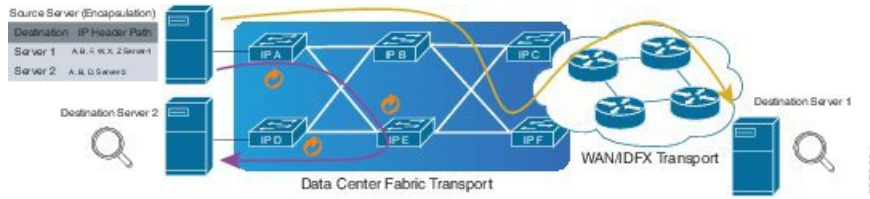
IP-in-IP encapsulation involves the insertion of an outer IP header over the existing IP header. The source and destination address in the outer IP header point to the end points of the IP-in-IP tunnel. The stack of IP headers are used to direct the packet over a predetermined path to the destination, provided the network administrator knows the loopback addresses of the routers transporting the packet. This tunneling mechanism can be used for determining availability and latency for most network architectures. It is to be noted that the entire path from source to the destination does not have to be included in the headers, but a segment of the network can be chosen for directing the packets.



Note The router only supports decapsulation and no encapsulation. Encapsulation is done by remote routers.

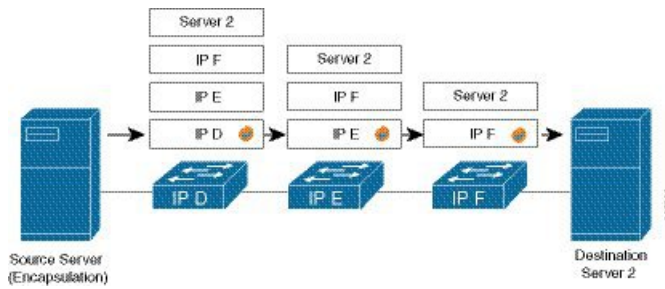
The following topology describes a use case where IP-in-IP encapsulation and de-capsulation is used for different segments of the network from source to destination. The IP-in-IP tunnel consists of multiple routers used to de-capsulate and direct the packet through the data center fabric network.

Figure 1: IP-in-IP Decapsulation through a Data Center Network



The following illustration shows how the stacked IPv4 headers are de-capsulated as they traverse through the de-capsulating routers.

Figure 2: IP Header De-capsulation



Stacked IP Header in an Encapsulated Packet

The encapsulated packet will have an outer IPv4 header stacked over the original IPv4 header, as shown in the following illustration.

Figure 3: Encapsulated Packet

[-] Frame	
[-] EthernetII	
Preamble (hex)	fb555555555555d5
Destination MAC	62:19:88:64:E2:68
Source MAC	00:10:94:00:00:02
EtherType (hex)	<auto> Internet IP
[-] IPv4 Header	
Version (int)	<auto> 4
Header length (int)	<auto> 5
ToS/DiffServ	tos (0x00)
Total length (int)	<auto> calculated
Identification (int)	0
[-] Control Flags	
Reserved (bit)	0
DF Bit (bit)	0
MF Bit (bit)	0
Fragment Offset (int)	0
Time to live (int)	255
Protocol (int)	<auto> IP
Checksum (int)	<auto> 33492
Source	192.xx.xx.xx
Destination	127.0.0.1
Header Options	
Gateway	192.0.2.10
[-] IPv4 Header	
Version (int)	<auto> 4
Header length (int)	<auto> 5
ToS/DiffServ	tos (0x00)
Total length (int)	<auto> calculated
Identification (int)	0
[-] Control Flags	
Reserved (bit)	0

385413

Configuration

You can use the following sample configuration on the routers to decapsulate the packet as it traverses the IP-in-IP tunnel:

```
Router(config)# interface loopback 0
Router(config-if)# ipv4 address 127.0.0.1/32
Router(config-if)# no shutdown
Router(config-if)# interface tunnel-ip 10
```

```
Router(config-if)# ipv4 unnumbered loopback 1
Router(config-if)# tunnel mode ipv4 decap
Router(config-if)# tunnel source loopback 0
```

- **tunnel-ip**: configures an IP-in-IP tunnel interface.
- **ipv4 unnumbered loopback address**: enables ipv4 packet processing without an explicit address, except for loopback address.
- **tunnel mode ipv4 decap**: enables IP-in-IP de-capsulation.
- **tunnel source**: indicates the source address for the IP-in-IP decap tunnel with respect to the router interface.



Note You can configure the tunnel destination only if you want to decapsulate packets from a particular destination. If no tunnel destination is configured, then all the ip-in-ip ingress packets on the configured interface are decapsulated.

Running Configuration

```
Router# show running-config interface tunnel-ip 10
...
interface tunnel-ip 10
ipv4 unnumbered loopback 1
tunnel mode ipv4 decap
```

Extended ACL needs to match on the outer header for IP-in-IP De-capsulation

Starting with Cisco IOS XR Software Release 7.0.14, extended ACL has to match on the outer header for IP-in-IP De-capsulation. Extended ACL support reduces mirrored traffic throughput. This match is based only on the IPv4 protocol, and extended ACL is applied to the received outermost IP header, even if the outer header is locally terminated.

Sample configuration:

```
RP/0/RP0/CPU0:router#show running-config interface bundle-Ether 50.5
Tue May 26 12:11:49.017 UTC
interface Bundle-Ether50.5
ipv4 address 101.1.5.1 255.255.255.0
encapsulation dot1q 5
ipv4 access-group ExtACL_IPinIP ingress
ipv4 access-group any_dscpegg egress
!
```

```
RP/0/RP0/CPU0:router#show access-lists ipv4 ExtACL_IPinIP hardware ingress location$
Tue May 26 12:11:55.940 UTC
ipv4 access-list ExtACL_IPinIP
10 permit ipv4 192.168.0.0 0.0.255.255 any ttl gt 150
11 deny ipv4 172.16.0.0 0.0.255.255 any fragments
12 permit ipv4 any any
```

Controlling the TTL Value of Inner Payload Header

Cisco 8000 Routers allow you to control the TTL value of inner payload header of IP-in-IP tunnel packets before it gets forwarded to the next-hop router. This feature enables a router to forward custom formed IP-in-IP stacked packets even if the inner packet TTL is 1. Therefore, this feature enables you to measure the link-state and path reachability from end to end in a network.



Note After you enable or disable the decrement of the TTL value of the inner payload header of a packet, you do not need to reload the line card.

Configuration

To disable the decrement of the TTL value of inner payload header of an IP-in-IP packet, use the following steps:

1. Enter the global configuration mode.
2. Disable the decrement of TTL value of inner payload header of an IP-in-IP packet.

Configuration Example

```
/* Enter the Global Configuration mode. */
Router# configure

/* Disable the decrement of TTL value of inner payload header of an IP-in-IP packet. */
Router(config)# hw-module profile cef ttl tunnel-ip decrement disable
Router(config)# commit
```

Associated Commands

- [hw-module profile cef ttl tunnel-ip decrement disable](#)

