



# CHAPTER 7

## Configuring Load Balancing on the Home Agent

---

This chapter discusses concepts and configuration details regarding Server Load Balancing on the Cisco Mobile Wireless Home Agent.

This chapter includes the following sections:

- [HA Server Load Balancing, page 7-1](#)
- [Load Balancing in HA-SLB, page 7-3](#)
- [HA-SLB Operating Modes, page 7-3](#)
- [Configuring HA Load Balancing, page 7-3](#)
- [Configuring Server Load Balancing, page 7-3](#)
- [HA-SLB Configuration Examples, page 7-4](#)

### HA Server Load Balancing



**Note**

---

The HA-Server Load Balancing (HA-SLB) feature is not available on the Cisco 7301 Series Router

---

The HA-SLB feature is built upon the existing IOS Server Load Balancing (SLB) feature. SLB allows users to represent a group of network servers (a server farm) as a single server instance, balance the traffic to the servers, and limit traffic to individual servers. The single server instance that represents a server farm is referred to as a virtual server. The servers that comprise the server farm are referred to as real servers.

SLB can distribute the traffic to real servers through mechanisms like round robin to real servers. Additionally, it can monitor the health of each real server using the Dynamic Feedback Protocol, choose a server that has the least load, and choose a server that is up and running. Please refer to the following URL for more information on SLB architecture:

[http://www.cisco.com/en/US/products/ps5940/products\\_white\\_paper0900aecd802921f0.shtml](http://www.cisco.com/en/US/products/ps5940/products_white_paper0900aecd802921f0.shtml)

The HA-SLB feature is available on the Cisco 7600 series platforms. This feature allows a set of real Home Agents, each running on an SAMI, to be identified by a single virtual server IP address residing on the Cisco 7600 Supervisor.

PDSN/FAs send an initial registration request for a user to the virtual server IP address. HA-SLB running on the SUP intercepts the packets and forwards the registration request to one of the real Home Agents.

A typical call flow would have the following sequence of events:

- 
- Step 1** The PDSN/FA forwards a Mobile IP RRQ to virtual server IP address (HA-SLB). If the AAA server returns the HA address to the PDSN/FA, the AAA server must be configured to return the address of virtual server IP address.
- Step 2** SLB picks one of the real server/HAs from its serverfarm and it delivers Mobile IP RRQ to this server.
- Step 3** The real HA responds to MobileIP RRQ with a Reply, the message is sent from the real HA to the PDSN/FA. The HA-SLB does not intercept this packet. The real HA creates binding and local tunnel endpoint.
- Step 4** The PDSN/FA creates a visitor table entry and local tunnel endpoint, and sends/receives traffic through the tunnel directly from Real HA
- Step 5** The PDSN/FA sends a Mobile IP RRQ with lifetime of “0” to the real HA to close the binding.




---

**Note** Note that the packet is not sent to virtual IP address (HA-SLB)

---

- Step 6** The Real HA sends Mobile IP RRP to the PDSN/FA. The HA-SLB does not intercept this packet. The Real HA closes the binding.



**Note**

---

The Mobile IP Messages are not compliant with RFC 2002. But they are compliant to draft-kulkarni-mobile-ip-dynamic-ha-assignment-frmrwrk-00.txt.

---

RRQs destined to the HA/SLB virtual IP address, with an HA address of 0.0.0.0 or 255.255.255.255, are forwarded to the actual HA using a weighted “round-robin,” load balancing algorithm. The SLB mechanism supports Dynamic Feedback Protocol (DFP) that gives real servers the ability to communicate real server health to the load balancer, thereby adjusting the weight of the real server in the load balancing algorithms.

Since the MN can send multiple RRQs before it hears a RRP from the HA (either the MN power cycles after sending an initial RRQ, or it is mis-configured to send multiple initial registrations, or RRP are dropped by the network), it is important to keep track of registrations coming from the same MN. This avoids the case where the same MN is registered at multiple HAs, and wastes IP addresses and other resources at those HAs. To solve this problem, HA-SLB would parse the RRQ and create a session object indexed by the MNs NAI. This session object will store the real HA IP address where the RRQ was forwarded. Subsequent registrations from the same MN will be forwarded to this same real HA. The session object will be stored for a configurable period of time (default to 10 seconds). If the HA-SLB does not see a RRQ from the MN within this period of time, the session object is cleared. If HA-SLB sees a RRQ, the timer associated with the session object is reset.

A retry counter is associated with each session object, and is incremented for each re-transmitted RRQ seen by the load balancer. If the number of retries seen is greater than the configured “reassign” threshold, the session sending the retransmissions will be re-assigned to another real HA, and a connection failure is recorded for the original real HA. Real servers are assumed to be down and no more RRQs re-directed to them when enough connection failures are seen to reach a configured threshold. HA-SLB will restart directing sessions to that real server after a configurable time interval or if the real server sends a DFP message to HA-SLB.

---

## Load Balancing in HA-SLB

HA-SLB uses a weighted round-robin load-balancing algorithm. This algorithm specifies that the real server used for a new connection to the virtual server is chosen from the server farm in a circular fashion. Each real server is assigned a weight  $n$ , that represents its capacity to handle connections, as compared to the other real servers associated with the virtual server. As an example, assume a server farm comprised of real server ServerA with  $n = 3$ , ServerB with  $n = 1$ , and ServerC with  $n = 2$ . The first three RRQs to the virtual server are assigned to ServerA, the fourth RRQ to ServerB, and the fifth and sixth RRQs to ServerC.

It is possible to configure IOS SLB for either static or dynamic load balancing. Static load balancing is achieved by assigning weights statically to each HA in the server farm. Dynamic load balancing is achieved by configuring Dynamic Feedback Protocol (DFP), with the DFP manager on SLB, and the DFP client on each of the real HAs.

## HA-SLB Operating Modes

HA-SLB operates in two modes, Dispatched mode and Direct (NAT server) mode.

In Dispatched mode the virtual server address is known to the HAs. HA-SLB will simply redirect packets to the HAs at the MAC layer. This requires the HAs to be layer 2 adjacent to SLB.

In Direct mode, HA-SLB works in NAT server mode and routes the RRQs to the HAs by changing the destination IP address in the RRQ to that of the real server. As a result the HAs need not be layer 2 adjacent to SLB.

To configure your routers for Mobile IP HA redundancy, perform the required tasks described in the following sections:

- [Configuring HA Load Balancing, page 7-3](#)
- [Configuring Server Load Balancing, page 7-3](#)

## Configuring HA Load Balancing

To enable the HA Load Balancing feature, perform these tasks:

	Command	Purpose
Step 1	Router(config)# <b>ip mobile home-agent dynamic-address</b> <i>ip address</i>	Sets the Home Agent Address field in the Registration Response packet. The Home Agent Address field will be set to <i>ip address</i> . This command is configured on the HA.

## Configuring Server Load Balancing

To enable the Mobile IP SLB feature on the HA, perform the following task:

	Command	Purpose
Step 1	Router(config)# <b>ip slb vservers</b> <i>name</i> Router(config-slb-vservers)# <b>virtual</b> <i>ip address</i> <b>udp</b> <b>434 service</b> <i>ipmobile</i>	Enables the Mobile IP SLB feature. The <i>ip address</i> is the virtual Home Agent address to which registration requests from PDSN/FA will be sent. This is configured on the SLB Supervisor.

## HA-SLB Configuration Examples

The following examples illustrate various HA-SLB configurations, including how to verify details of the configurations.

### Dispatched MODE WITH STATIC WEIGHTS

#### Configuration on SLB:

The following commands configure a serverfarm “HAFARM”, and associate two real servers (HAs) with the serverfarm. The real servers are configured with a static weight of one.

```
ip slb serverfarm HAFARM
  real 10.1.1.51
    weight 1
  inservice
!
  real 10.1.1.52
    weight 1
  inservice
```

The following commands configure a virtual server with service as “ipmobile” on the SLB and associates the serverfarm “HAFARM” with the virtual server. Optionally, the **idle ipmobile request** *idle-time-val* command configures the duration for which the session object exists.

```
ip slb vserver MIPS LB
  virtual 10.1.1.10 udp 434 service ipmobile
  serverfarm HAFARM
  idle ipmobile request 300
  inservice
```

#### Configuration on HA:

The following command configures the virtual server address as a loopback address on the HA. This configuration is required only for Dispatched mode.

```
interface Loopback1
ip address 10.1.1.10 255.255.255.0
```

The following command sets the source address and HA address field in the RRP to that of the real HA’s address. This configuration is required only for Dispatched mode.

```
ip mobile home-agent dynamic-address 10.1.1.51
```

#### Show Output on SLB:

The following command displays the status of server farm “HAFARM” and, the associated real servers, and their status. It also shows the number of connections assigned to each of the real servers.

The show output below was captured after opening 4 MIP sessions which HA-SLB has load balanced equally across two real HA's (2 connections to each HA).

```
SLB-7600#show ip slb reals
```

real	farm name	weight	state	conns
20.1.1.51	HAFARM	1	OPERATIONAL	2
20.1.1.52	HAFARM	1	OPERATIONAL	2

The following command displays all the sessions during runtime, or as long as the session objects exist.

```
SLB-7600#show ip slb sessions ipmobile
```

vserver	NAI hash	client	real	state
MIPSLB	A984DF0A00000000	15.1.1.51	20.1.1.52	IPMOBILE_ESTAB
MIPSLB	1DC0E31400000000	15.1.1.51	20.1.1.52	IPMOBILE_ESTAB
MIPSLB	2BDEE91100000000	15.1.1.51	20.1.1.51	IPMOBILE_ESTAB
MIPSLB	47E2FD1B00000000	15.1.1.51	20.1.1.51	IPMOBILE_ESTAB

### Show Output on HAs:

The following command shows that two bindings each were opened on HA1 and HA2.

```
HA1-7600#show ip mobile binding summary
Mobility Binding List:
Total 2
HA1-7600#
```

```
HA2-7600#show ip mobile binding summary
Mobility Binding List:
Total 2
HA2-7600#
```

## Dispatched mode with DFP

### Configuration on SLB:

The following commands configure a serverfarm “HAFARM” and associates two real servers (HAs) with the serverfarm.

```
ip slb serverfarm HAFARM
  real 10.1.1.51
  inservice
!
  real 10.1.1.52
  inservice
!
```

The following commands configure a virtual server with service as “ipmobile” on the SLB and associates the serverfam HAFARM with the virtual server. The optional config command below ‘idle ipmobile request *idle-time-val*’ configures the duration for which the session object exists.

```
ip slb vserver MIPSLB
  virtual 10.1.1.10 udp 434 service ipmobile
  serverfarm HAFARM
  idle ipmobile request 300
  inservice
```

The following command configures the DFP Manager on HA-SLB and assigns two DFP agents (clients) to which HA-SLB can connect to.

```
ip slb dfp
  agent 10.1.1.51 500
  agent 10.1.1.52 500
!
```

**Configuration on HA:**

The following command configures the virtual server address as a loopback address on the HA. This configuration is required only for Dispatched mode.

```
interface Loopback1
ip address 10.1.1.10 255.255.255.0
!
```

The following command configures the DFP agent on the real HA. The port num. configured here must match the port number specified on the DFP Manager.

```
ip dfp agent ipmobile
port 500
inservice
!
```

The following command sets the source address and HA address field in the RRP to that of the real HA's address. This config is required only for Dispatched mode.

```
ip mobile home-agent dynamic-address 10.1.1.51
```

**Show Output on SLB:**

The following command verifies that the HAs report an initial weight of 25 (default weight) when DFP is configured.

```
SLB-7600#show ip slb dfp weights
Real IP Address: 10.1.1.51 Protocol: UDP Port: 434 Bind_ID: 65535 Weight: 25
Set by Agent 10.1.1.51:500 at 14:59:23 UTC 04/21/03
Real IP Address: 10.1.1.52 Protocol: UDP Port: 434 Bind_ID: 65535 Weight: 25
Set by Agent 10.1.1.52:500 at 14:59:15 UTC 04/21/03
SLB-7600#
```

The following command displays the status of server farm HAFARM and, the associated real servers, and their status. It also shows the no. of connections assigned to each of the real servers.

The show output below was captured after opening 100 MIP sessions which HA-SLB has load balanced equally across two real HA's (50 connections to each HA).

```
SLB-7600#show ip slb reals
```

real	farm name	weight	state	conns
10.1.1.51	HAFARM	24	OPERATIONAL	50
10.1.1.52	HAFARM	24	OPERATIONAL	50

```
SLB-7600#
```

**Show output on HAs:**

The following command verifies that 50 bindings each were opened on HA1 and HA2

```
HA1-7600#show ip mobile binding summary
Mobility Binding List:
Total 50
HA1-7600#
```

```
HA2-7600#show ip mobile binding summary
Mobility Binding List:
Total 50
HA2-7600#
```

Currently, the number of bindings and memory usage are considered for calculating the load balancing in HA-SLB. The existing DFP (dynamic feedback protocol) weight calculation equation can be modified by considering the CPS (frequency of calls per second), and throughput parameters on each real server (HA).

The CPS on the HA calculated for every minute is called the Usage CPS, and can be configured to some maximum value (Available CPS) that can be handled by the HA. If the Usage CPS reaches the available CPS then the HA real server will return less weight to the SLB.

It is difficult to calculate throughput on a router, and it can be solved by usage of interrupt CPU for packet handling.

From the above two parameters, the equation looks like,

$$\text{dfp\_weight} = (\text{Maxbindings} - \text{NumberofBindings}) * (\text{cpu} + \text{mem}) * \\ (\text{Available cps} - \text{Usage cps}) * \text{dfp\_max\_weight} / (\text{Maxbindings} * 32 * \text{Available cps})$$



**Note**

Currently a MIB item that contains metrics is not available.

## Direct Mode With Static Weights

### Configuration on SLB:

The following commands configure a serverfarm “HAFARM” and associates two real servers (HAs) with the serverfarm. The real servers are configured with a static weight of one. The command **nat server** configures HA-SLB in Direct (Nat server) mode of operation.

```
ip slb serverfarm HAFARM
nat server
real 10.1.1.51
  weight 1
  inservice
!
real 10.1.1.52
  weight 1
  inservice

ip slb vserver MIPS LB
virtual 10.1.1.10 udp 434 service ipmobile
serverfarm HAFARM
idle ipmobile request 300
inservice
```

### Show Output on SLB:

The following example displays the status of server farm HAFARM, the associated real servers, and their status. It also shows the number of connections assigned to each of the real servers.

The show output below was captured after opening 4 MIP sessions which the HA-SLB load balanced equally across two real HAs (2 connections to each HA).

```
SLB-7600#show ip slb reals
```

real	farm name	weight	state	conns
10.1.1.51	HAFARM	1	OPERATIONAL	2
10.1.1.52	HAFARM	1	OPERATIONAL	2

The following command display all the sessions during runtime, or as long as the session objects exist.

```
SLB-7600#show ip slb sessions ipmobile

vserver          NAI hash          client            real              state
-----
MIPSLB           A984DF0A00000000 15.1.1.51        10.1.1.52        IPMOBILE_ESTAB
MIPSLB           1DC0E31400000000 15.1.1.51        10.1.1.52        IPMOBILE_ESTAB
MIPSLB           2BDEE91100000000 15.1.1.51        10.1.1.51        IPMOBILE_ESTAB
MIPSLB           47E2FD1B00000000 15.1.1.51        10.1.1.51        IPMOBILE_ESTAB

SLB-7600#
```

### Show Output on HAs:

The following example shows that 2 bindings each were opened on HA1 and HA2.

```
HA1-7600#show ip mobile binding summary
Mobility Binding List:
Total 2
HA1-7600#

HA2-7600#show ip mobile binding summary
Mobility Binding List:
Total 2
HA2-7600#
```

The following debug when enabled shows NAT server mode is operational:

```
SLB-7600#debug ip slb sessions ipmobile
SLB-7600#
*Apr 21 15:25:58: %SYS-5-CONFIG_I: Configured from console by console
*Apr 21 15:26:03: SLB_SESSION_IPMOBILE: client = 15.1.1.51, NAI:
mwtS-mip-np-user1@ispxyz.com, length: 28
*Apr 21 15:26:03: SLB_SESSION_IPMOBILE: event= IPMOBILE_REQ_REQUEST, state= IPMOBILE_INIT
-> IPMOBILE_ESTAB
*Apr 21 15:26:03: SLB_SESSION: v_ip= 15.1.1.10:434 ( 7), real= 10.1.1.51, NAT= S
*Apr 21 15:26:03: SLB_SESSION: client= 15.1.1.51:434 session_key= 47E2FD1B00000000
SLB-7600#
```

## Direct Mode with DFP

### Configuration on SLB:

The following commands configure a serverfarm “HAFARM” and associates two real servers (HAs) with the serverfarm. The **nat server** command configures HA-SLB in Direct (Nat server) mode of operation.

```
ip slb serverfarm HAFARM
nat server
real 10.1.1.51
  inservice
!
real 10.1.1.52
  weight 1
  inservice
!
```

The following commands configure a virtual server with service as “ipmobile” on the SLB and associates the serverfarm HAFARM with the virtual server. The optional **idle ipmobile request idle-time-val** config command configures the duration for which the session object exists.



```
ip slb vserver MIPSLB
  virtual 10.1.1.10 udp 434 service ipmobile
  serverfarm HAFARM
  idle ipmobile request 300
  inservice
!
```

The following command configures the DFP Manager on HA-SLB and assigns two DFP agents (clients) to which HA-SLB can connect to.

```
ip slb dfp
  agent 10.1.1.51 500
  agent 10.1.1.52 500
```

#### Configuration on HA:

The following command configures the DFP agent on the real HA. The port number that is configured must match the port number specified on the DFP Manager.

```
ip dfp agent ipmobile
  port 500
  inservice
!
```

#### Show Output on SLB:

The following command verifies that the HAs report an initial weight of 25 (default weight) when DFP is configured.

```
SLB-7600#show ip slb dfp weights
  Real IP Address: 10.1.1.51 Protocol: UDP Port: 434 Bind_ID: 65535 Weight: 25
    Set by Agent 10.1.1.51:500 at 14:59:23 UTC 04/21/03
  Real IP Address: 10.1.1.52 Protocol: UDP Port: 434 Bind_ID: 65535 Weight: 25
    Set by Agent 10.1.1.52:500 at 14:59:15 UTC 04/21/03
SLB-7600#
```

The following command displays the status of server farm “HAFARM”, the associated real servers, and their status. It also shows the number of connections assigned to each of the real servers.

The show output below was captured after opening 100 MIP sessions which HA-SLB has load balanced equally across two real HAs (50 connections to each HA).

```
SLB-7600#show ip slb reals
```

real	farm name	weight	state	conns
10.1.1.51	HAFARM	24	OPERATIONAL	50
10.1.1.52	HAFARM	24	OPERATIONAL	50

```
SLB-7600#
```

#### Show Output on HAs:

The following command shows that 50 bindings each were opened on HA1 and HA2.

```
HA1-7600#show ip mobile binding summary
Mobility Binding List:
Total 50
HA1-7600#
```

```
HA2-7600#show ip mobile binding summary
Mobility Binding List:
Total 50
HA2-7600#
```

The following debug when enabled shows NAT server mode is operational:

```
SLB-7600#debug ip slb sessions ipmobile
SLB-7600#
*Apr 21 15:47:16: SLB_SESSION_IPMOBILE: client = 10.1.1.51, NAI:
mwtS-mip-np-user1@ispxyz.com, length: 28
*Apr 21 15:47:16: SLB_SESSION_IPMOBILE: event= IPMOBILE_REQ_REQUEST, state= IPMOBILE_INIT
-> IPMOBILE_ESTAB
*Apr 21 15:47:16: SLB_SESSION: v_ip= 10.1.1.10:434 ( 7), real= 20.1.1.51, NAT= S
*Apr 21 15:47:16: SLB_SESSION: client= 10.1.1.51:434 session_key= 47E2FD1B00000000
*Apr 21 15:47:16: SLB_SESSION_IPMOBILE: client = 15.1.1.51, NAI:
mwtS-mip-np-user2@ispxyz.com, length: 28
*Apr 21 15:47:16: SLB_SESSION_IPMOBILE: event= IPMOBILE_REQ_REQUEST, state= IPMOBILE_INIT
-> IPMOBILE_ESTAB
*Apr 21 15:47:16: SLB_SESSION: v_ip= 10.1.1.10:434 ( 7), real= 20.1.1.51, NAT= S
*Apr 21 15:47:16: SLB_SESSION: client= 10.1.1.51:434 session_key= 1DC0E31400000000
```

## Direct Mode of Operation and Crypto Transform Mode is Tunnel

```
Configuration on SLB:
ip slb serverfarm FARM1
  nat server
  real 10.99.11.11
  inservice
!
  real 10.99.11.12
  inservice
!
ip slb vserver IPSECSLB
  virtual 15.1.1.10 udp 434 service ipmobile
  serverfarm FARM1
  inservice
```

The following commands configure IPSEC on HA-SLB:

```
crypto isakmp policy 1
  authentication pre-share
crypto isakmp key cisco address 10.1.1.51
!
!
crypto ipsec transform-set esp-des-sha-transport ah-sha-hmac esp-des
!
crypto map l2tpmap 10 ipsec-isakmp
  set peer 10.1.1.51
  set transform-set esp-des-sha-transport
  match address 101
!
interface GigabitEthernet6/1 (inside port of the IPSEC module)
  no ip address
  switchport
  switchport trunk encapsulation dot1q
  switchport trunk allowed vlan 1,15,1002-1005
  switchport mode trunk
  cdp enable
!
interface GigabitEthernet6/2 (outside port of the IPSEC module)
  no ip address
  switchport
  switchport trunk encapsulation dot1q
  switchport trunk allowed vlan 1,16,1002-1005
```

```

switchport mode trunk
cdp enable
!
interface FastEthernet3/15
no ip address
duplex full
speed 100
crypto connect vlan 15
!
!
interface Vlan15
ip address 10.1.1.15 255.0.0.0
no ip redirects
no ip unreachable
no mop enabled
crypto map l2tpmap
!
!
access-list 101 permit ip host 10.1.1.10 host 10.1.1.51

```

### Configuration on PDSN:

The following commands configure IPSEC on PDSN:

```

crypto isakmp policy 1
authentication pre-share
crypto isakmp key cisco address 10.1.1.15
!
!
crypto ipsec transform-set esp-des-sha-transport esp-des esp-sha-hmac
!
crypto map l2tpmap 10 ipsec-isakmp
set peer 10.1.1.15
set transform-set esp-des-sha-transport
match address 101

interface FastEthernet1/0
ip address 10.1.1.51 255.0.0.0
duplex full
crypto map l2tpmap

access-list 101 permit ip host 10.1.1.51 host 10.1.1.10

```

Execute **clear crypto isakmp** and **clear crypto sa** on the PDSN and SLB. Open multiple MIP flows.

Show Output on PDSN:

The following command is used to verify that packets sent out of PDSN are encrypted:

```
PDSN-7600#sh crypto ipsec sa
```

```

interface: FastEthernet1/0
  Crypto map tag: l2tpmap, local addr. 10.1.1.51

local ident (addr/mask/prot/port): (10.1.1.51/255.255.255.255/0/0)
remote ident (addr/mask/prot/port): (10.1.1.10/255.255.255.255/0/0)
current_peer: 10.1.1.15
  PERMIT, flags={origin_is_acl,}
  #pkts encaps: 4, #pkts encrypt: 4, #pkts digest 4
  #pkts decaps: 0, #pkts decrypt: 0, #pkts verify 0
  #pkts compressed: 0, #pkts decompressed: 0
  #pkts not compressed: 0, #pkts compr. failed: 0, #pkts decompress failed: 0
  #send errors 4, #recv errors 0

local crypto endpt.: 10.1.1.51, remote crypto endpt.: 10.1.1.15

```

```

path mtu 1500, media mtu 1500
current outbound spi: 1A274E9D

inbound esp sas:
spi: 0xD3D5F08B(3554013323)
  transform: esp-des ,
  in use settings =(Tunnel, )
  slot: 0, conn id: 2002, flow_id: 1, crypto map: l2tpmap
  sa timing: remaining key lifetime (k/sec): (4608000/3026)
  IV size: 8 bytes
  replay detection support: Y

inbound ah sas:
spi: 0x7FEE86C3(2146338499)
  transform: ah-sha-hmac ,
  in use settings =(Tunnel, )
  slot: 0, conn id: 2000, flow_id: 1, crypto map: l2tpmap
  sa timing: remaining key lifetime (k/sec): (4608000/3026)
  replay detection support: Y

inbound pcp sas:

outbound esp sas:
spi: 0x1A274E9D(438783645)
  transform: esp-des ,
  in use settings =(Tunnel, )
  slot: 0, conn id: 2003, flow_id: 2, crypto map: l2tpmap
  sa timing: remaining key lifetime (k/sec): (4607999/3026)
  IV size: 8 bytes
  replay detection support: Y

outbound ah sas:
spi: 0x5F9A83(6265475)
  transform: ah-sha-hmac ,
  in use settings =(Tunnel, )
  slot: 0, conn id: 2001, flow_id: 2, crypto map: l2tpmap
  sa timing: remaining key lifetime (k/sec): (4607999/3026)
  replay detection support: Y

outbound pcp sas:

```

```
PDSN-7600#
```

### Show Output on SLB:

The following command is used to verify that packets received by HA-SLB are decrypted:

```
SLB1-7600#sh crypto ipsec sa
```

```

interface: Vlan15
  Crypto map tag: l2tpmap, local addr. 10.1.1.15

local ident (addr/mask/prot/port): (10.1.1.10/255.255.255.255/0/0)
remote ident (addr/mask/prot/port): (10.1.1.51/255.255.255.255/0/0)
current_peer: 15.1.1.51
  PERMIT, flags={origin_is_acl,}
  #pkts encaps: 0, #pkts encrypt: 0, #pkts digest 0
  #pkts decaps: 4, #pkts decrypt: 4, #pkts verify 0
  #pkts compressed: 0, #pkts decompressed: 0
  #pkts not compressed: 0, #pkts compr. failed: 0, #pkts decompress failed: 0
  #send errors 0, #recv errors 0

```

```

local crypto endpt.: 15.1.1.15, remote crypto endpt.: 10.1.1.51
path mtu 1500, media mtu 1500
current outbound spi: D6C550E1

```

```

inbound esp sas:
spi: 0x267FCD46(645909830)
transform: esp-des ,
in use settings ={Tunnel, }
slot: 0, conn id: 11027, flow_id: 63, crypto map: 12tpmap
sa timing: remaining key lifetime (k/sec): (4607999/3581)
IV size: 8 bytes
replay detection support: Y

```

```

inbound ah sas:
spi: 0xF779A01E(4151943198)
transform: ah-sha-hmac ,
in use settings ={Tunnel, }
slot: 0, conn id: 11025, flow_id: 63, crypto map: 12tpmap
sa timing: remaining key lifetime (k/sec): (4607999/3581)
replay detection support: Y

```

```
inbound pcp sas:
```

```

outbound esp sas:
spi: 0xD6C550E1(3603255521)
transform: esp-des ,
in use settings ={Tunnel, }
slot: 0, conn id: 11028, flow_id: 64, crypto map: 12tpmap
sa timing: remaining key lifetime (k/sec): (4608000/3581)
IV size: 8 bytes
replay detection support: Y

```

```

outbound ah sas:
spi: 0x325BEB84(844884868)
transform: ah-sha-hmac ,
in use settings ={Tunnel, }
slot: 0, conn id: 11026, flow_id: 64, crypto map: 12tpmap
sa timing: remaining key lifetime (k/sec): (4608000/3581)
replay detection support: Y

```

```
outbound pcp sas:
```

```
SLB1-7600#sh ip slb sessions ipmobile
```

vserver	NAI hash	client	real	state
IPSECSLB	A984DFOA00000000	10.1.1.51	10.99.11.12	IPMOBILE_ESTAB
IPSECSLB	1DC0E31400000000	10.1.1.51	10.99.11.12	IPMOBILE_ESTAB
IPSECSLB	2BDEE91100000000	10.1.1.51	10.99.11.11	IPMOBILE_ESTAB
IPSECSLB	47E2FD1B00000000	10.1.1.51	10.99.11.11	IPMOBILE_ESTAB

```
SLB1-7600#
```

```
SLB1-7600#sh ip slb
```

```
SLB1-7600#sh ip slb rea
```

```
SLB1-7600#sh ip slb reals
```

real	farm name	weight	state	conns
10.99.11.11	FARM1	1	OPERATIONAL	2
10.99.11.12	FARM1	1	OPERATIONAL	2

```
SLB1-7600
```

```
Show output on SLB:
```

```
HA5-2#sh ip mob binding summary
```

```
Mobility Binding List:
```

```
Total 2
HA5-2#

HA5-3#sh ip mob binding summary
Mobility Binding List:
Total 2
HA5-3#
```

**Debug Output on SLB:**

The following debug when enabled shows NAT server mode is operational:

```
SLB1-7600#debug ip slb sessions ipmobile
*Jul 1 05:25:25.513: SLB_SESSION_IPMOBILE: event= IPMOBILE_TIMEOUT, state= IPMOBILE_ESTAB
-> IPMOBILE_INIT
*Jul 1 05:25:25.513: SLB_SESSION: v_ip= 15.1.1.10:434 ( 7), real= 99.99.11.12, NAT= S
*Jul 1 05:25:25.513: SLB_SESSION: client= 15.1.1.51:434 session_key= A984DF0A00000000
*Jul 1 05:25:25.513: SLB_SESSION_IPMOBILE: event= IPMOBILE_TIMEOUT, state= IPMOBILE_ESTAB
-> IPMOBILE_INIT
*Jul 1 05:25:25.513: SLB_SESSION: v_ip= 15.1.1.10:434 ( 7), real= 99.99.11.11, NAT= S
*Jul 1 05:25:25.513: SLB_SESSION: client= 15.1.1.51:434 session_key= 2BDEE91100000000
*Jul 1 05:25:25.513: SLB_SESSION_IPMOBILE: event= IPMOBILE_TIMEOUT, state= IPMOBILE_ESTAB
-> IPMOBILE_INIT
```

**Direct Mode of Operation and Crypto Transform Mode is Transport****Configuration on SLB:**

```
ip slb serverfarm FARM1
  nat server
  real 10.99.11.11
  inservice
  !
  real 10.99.11.12
  inservice
  !
ip slb vserver IPSECSLB
  virtual 10.1.1.10 udp 434 service ipmobile
  serverfarm FARM1
  inservice
```

The following commands configure IPSEC on HA-SLB:

```
crypto isakmp policy 1
  authentication pre-share
crypto isakmp key cisco address 10.1.1.51
!
!
crypto ipsec transform-set esp-des-sha-transport ah-sha-hmac esp-des
  mode transport (The crypto mode is configured as transport )
!
crypto map l2tpmap 10 ipsec-isakmp
  set peer 10.1.1.51
  set transform-set esp-des-sha-transport
  match address 101
!
interface GigabitEthernet6/1 (inside port of the IPSEC module)
  no ip address
  switchport
  switchport trunk encapsulation dot1q
  switchport trunk allowed vlan 1,15,1002-1005
  switchport mode trunk
```

```

    cdp enable
    !
interface GigabitEthernet6/2      (outside port of the IPSEC module)
  no ip address
  switchport
  switchport trunk encapsulation dot1q
  switchport trunk allowed vlan 1,16,1002-1005
  switchport mode trunk
  cdp enable
  !
interface FastEthernet3/15
  no ip address
  duplex full
  speed 100
  crypto connect vlan 15
  !
  !
interface Vlan15
  ip address 15.1.1.15 255.0.0.0
  no ip redirects
  no ip unreachable
  no mop enabled
  crypto map l2tpmap
  !
  !
access-list 101 permit ip host 15.1.1.10 host 15.1.1.51

```

### Configuration on PDSN:

The following commands configure IPSEC on PDSN:

```

crypto isakmp policy 1
  authentication pre-share
crypto isakmp key cisco address 10.1.1.15
!
!
crypto ipsec transform-set esp-des-sha-transport esp-des esp-sha-hmac
  mode transport      (The crypto mode is configured as transport )
!
crypto map l2tpmap 10 ipsec-isakmp
  set peer 10.1.1.15
  set transform-set esp-des-sha-transport
  match address 101

interface FastEthernet1/0
  ip address 10.1.1.51 255.0.0.0
  duplex full
  crypto map l2tpmap

access-list 101 permit ip host 15.1.1.51 host 15.1.1.10

```

Execute **clear crypto isakmp** and **clear crypto sa** on the PDSN and SLB. Open multiple MIP flows.

### Show Output on PDSN :

The following command is used to verify that packets sent out of PDSN are encrypted

```
PDSN-7600#sh crypto ipsec sa
```

```

interface: FastEthernet1/0
  Crypto map tag: l2tpmap, local addr. 10.1.1.51

    local ident (addr/mask/prot/port): (10.1.1.51/255.255.255.255/0/0)
    remote ident (addr/mask/prot/port): (10.1.1.10/255.255.255.255/0/0)

```

```

current_peer: 10.1.1.15
  PERMIT, flags={origin_is_acl,}
#pkts encaps: 4, #pkts encrypt: 4, #pkts digest 4
#pkts decaps: 0, #pkts decrypt: 0, #pkts verify 0
#pkts compressed: 0, #pkts decompressed: 0
#pkts not compressed: 0, #pkts compr. failed: 0, #pkts decompress failed: 0
#send errors 4, #recv errors 0

local crypto endpt.: 10.1.1.51, remote crypto endpt.: 10.1.1.15
path mtu 1500, media mtu 1500
current outbound spi: 6A0EBD82

inbound esp sas:
  spi: 0x13E0E556(333505878)
    transform: esp-des ,
    in use settings =(Tunnel, )
    slot: 0, conn id: 2002, flow_id: 1, crypto map: l2tpmap
    sa timing: remaining key lifetime (k/sec): (4608000/3535)
    IV size: 8 bytes
    replay detection support: Y

inbound ah sas:
  spi: 0xEFEEE153(4025409875)
    transform: ah-sha-hmac ,
    in use settings =(Tunnel, )
    slot: 0, conn id: 2000, flow_id: 1, crypto map: l2tpmap
    sa timing: remaining key lifetime (k/sec): (4608000/3535)
    replay detection support: Y

inbound pcp sas:

outbound esp sas:
  spi: 0x6A0EBD82(1779350914)
    transform: esp-des ,
    in use settings =(Tunnel, )
    slot: 0, conn id: 2003, flow_id: 2, crypto map: l2tpmap
    sa timing: remaining key lifetime (k/sec): (4607999/3535)
    IV size: 8 bytes
    replay detection support: Y

outbound ah sas:
  spi: 0x49BE92A3(1237226147)
    transform: ah-sha-hmac ,
    in use settings =(Tunnel, )
    slot: 0, conn id: 2001, flow_id: 2, crypto map: l2tpmap
    sa timing: remaining key lifetime (k/sec): (4607999/3535)
    replay detection support: Y

outbound pcp sas:

```

PDSN-7600#

### Show Output on SLB:

SLB1-7600#sh ip slb sessions ipmobile

vserver	NAI hash	client	real	state
IPSECSLB	A984DF0A00000000	10.1.1.51	99.99.11.12	IPMOBILE_ESTAB
IPSECSLB	1DC0E31400000000	10.1.1.51	99.99.11.12	IPMOBILE_ESTAB
IPSECSLB	2BDEE91100000000	10.1.1.51	99.99.11.11	IPMOBILE_ESTAB
IPSECSLB	47E2FD1B00000000	10.1.1.51	99.99.11.11	IPMOBILE_ESTAB



```

SLB1-7600#
SLB1-7600#sh ip slb rea
SLB1-7600#sh ip sib reals

real                farm name          weight  state          conns
-----
99.99.11.11         FARM1              1      OPERATIONAL    2
99.99.11.12         FARM1              1      OPERATIONAL    2
SLB1-7600#
SLB1-7600#

```

The following command is used to verify that packets received by HA-SLB are decrypted:

```

SLB1-7600#sh crypto ipsec sa

interface: Vlan15
  Crypto map tag: l2tpmap, local addr. 10.1.1.15

local ident (addr/mask/prot/port): (10.1.1.10/255.255.255.0/0)
remote ident (addr/mask/prot/port): (10.1.1.51/255.255.255.0/0)
current_peer: 10.1.1.51
  PERMIT, flags={origin_is_acl,}
  #pkts encaps: 0, #pkts encrypt: 0, #pkts digest 0
  #pkts decaps: 4, #pkts decrypt: 4, #pkts verify 0
  #pkts compressed: 0, #pkts decompressed: 0
  #pkts not compressed: 0, #pkts compr. failed: 0, #pkts decompress failed: 0
  #send errors 0, #recv errors 0

local crypto endpt.: 15.1.1.15, remote crypto endpt.: 15.1.1.51
path mtu 1500, media mtu 1500
current outbound spi: 13E0E556

inbound esp sas:
  spi: 0x6A0EBD82(1779350914)
    transform: esp-des ,
    in use settings ={Tunnel, }
    slot: 0, conn id: 11031, flow_id: 65, crypto map: l2tpmap
    sa timing: remaining key lifetime (k/sec): (4607999/3527)
    IV size: 8 bytes
    replay detection support: Y

inbound ah sas:
  spi: 0x49BE92A3(1237226147)
    transform: ah-sha-hmac ,
    in use settings ={Tunnel, }
    slot: 0, conn id: 11029, flow_id: 65, crypto map: l2tpmap
    sa timing: remaining key lifetime (k/sec): (4607999/3527)
    replay detection support: Y

inbound pcp sas:

outbound esp sas:
  spi: 0x13E0E556(333505878)
    transform: esp-des ,
    in use settings ={Tunnel, }
    slot: 0, conn id: 11032, flow_id: 66, crypto map: l2tpmap
    sa timing: remaining key lifetime (k/sec): (4608000/3527)
    IV size: 8 bytes
    replay detection support: Y

outbound ah sas:
  spi: 0xEFEEE153(4025409875)
    transform: ah-sha-hmac ,
    in use settings ={Tunnel, }

```

```
slot: 0, conn id: 11030, flow_id: 66, crypto map: l2tpmap
sa timing: remaining key lifetime (k/sec): (4608000/3524)
replay detection support: Y
```

```
outbound pcp sas:
```

```
SLB1-7600#
```

### Show Output on HA:

```
HA5-2#sh ip mob binding summary
Mobility Binding List:
Total 2
HA5-2#
```

```
HA5-3#sh ip mob binding summary
Mobility Binding List:
Total 2
HA5-3#
```