



Session Recovery

With robust hardware failover and redundancy protection, any hardware or software failures on the system can quickly be corrected. However, software failures can occur for numerous reasons, often without prior indication.

This chapter describes the Session Recovery feature that provides seamless failover and reconstruction of subscriber session information in the event of a hardware or software fault.



Important

Session Recovery is a licensed Cisco feature. A separate feature license may be required. Contact your Cisco account representative for detailed information on specific licensing requirements. For information on installing and verifying licenses, refer to the *Managing License Keys* section of *Software Management Operations*.

This chapter includes the following sections:

- [How Session Recovery Works, on page 1](#)
- [Additional ASR 5500 Hardware Requirements, on page 4](#)
- [Configuring the System to Support Session Recovery, on page 4](#)
- [Recovery Control Task Statistics, on page 8](#)

How Session Recovery Works

This section provides an overview of how this feature is implemented and the recovery process.

The Session Recovery feature provides seamless failover and reconstruction of subscriber session information in the event of a hardware or software fault within the system preventing a fully connected user session from being disconnected.

Session recovery is performed by mirroring key software processes (for example, session manager and AAA manager) within the system. These mirrored processes remain in an idle state (standby-mode) wherein they perform no processing, until they may be needed in the event of a software failure (for example, a session manager task aborts).

The system spawns new instances of "standby mode" session and AAA managers for each active control processor (CP) being used. These mirrored processes require both memory and processing resources, which means that additional hardware may be required to enable this feature (see [Additional ASR 5500 Hardware Requirements, on page 4](#)).

Other key system-level software tasks, such as VPN manager, are performed on a physically separate packet processing card to ensure that a double software fault (for example, session manager and VPN manager fails at same time on same card) cannot occur. The packet processing card that hosts the VPN manager process is in active mode and reserved by the operating system for this sole use when session recovery is enabled.

There are two modes of session recovery.

- **Task recovery mode:** Wherein one or more session manager failures occur and are recovered without the need to use resources on a standby packet processing card. In this mode, recovery is performed by using the mirrored "standby-mode" session manager task(s) running on active packet processing cards. The "standby-mode" task is renamed, made active, and is then populated using information from other tasks such as AAA manager. In case of Task failure, limited subscribers will be affected and will suffer outage only until the task starts back up.
- **Full packet processing card recovery mode:** Used when a packet processing card hardware failure occurs, or when a planned packet processing card migration fails. In this mode, the standby packet processing card is made active and the "standby-mode" session manager and AAA manager tasks on the newly activated packet processing card perform session recovery.

Session/Call state information is saved in the peer AAA manager task because each AAA manager and session manager task is paired together. These pairs are started on physically different packet processing cards to ensure task recovery.

There are some situations wherein session recovery may not operate properly. These include:

- Additional software or hardware failures occur during the session recovery operation. For example, an AAA manager fails while the state information it contained was being used to populate the newly activated session manager task.
- A lack of hardware resources (packet processing card memory and control processors) to support session recovery.



Important

After a session recovery operation, some statistics, such as those collected and maintained on a per manager basis (AAA Manager, Session Manager, etc.) are in general not recovered, only accounting and billing related information is checkpointed and recovered.

Session Recovery is available for the following functions:

- Any session needing L2TP LAC support (excluding regenerated PPP on top of an HA or GGSN session)
- ASR 5500 only – Closed RP PDSN services supporting simple IP, Mobile IP, and Proxy Mobile IP
- ASR 5500 only – eHRPD service (evolved High Rate Packet Data)
- ASR 5500 only – ePDG service (evolved Packet Data Gateway)
- GGSN services for IPv4 and PPP PDP contexts
- HA services supporting Mobile IP and/or Proxy Mobile IP session types with or without per-user Layer 3 tunnels
- ASR 5500 only – HNB-GW: HNB Session over IuH
- ASR 5500 only – HNB-GW: HNB-CN Session over IuPS and IuCS

- ASR 5500 only – HNB-GW: SeGW Session IPSec Tunnel
- ASR 5500 only – HSGW services for IPv4
- IPCF (Intelligent Policy Control Function)
- ASR 5500 only – IPSG-only systems (IP Services Gateway)
- LNS session types (L2TP Network Server)
- MME (Mobility Management Entity)
- ASR 5500 only – NEMO (Network Mobility)
- P-GW services for IPv4
- ASR 5500 only – PDIF (Packet Data Interworking Function)
- PDSN services supporting simple IP, Mobile IP, and Proxy Mobile IP
- S-GW (Serving Gateway)
- SaMOG (S2a Mobility over GTP) Gateway (CGW and MRME)
- ASR 5500 only – SAE-GW (System Architecture Evolution Gateway)
- ASR 5500 only – SGSN services (3G and 2.5G services) for IPv4 and PPP PDP contexts

Session recovery is **not supported** for the following functions:

- Destination-based accounting recovery
- GGSN network initiated connections
- GGSN session using more than 1 service instance
- MIP/L2TP with IPSec integration
- MIP session with multiple concurrent bindings
- Mobile IP sessions with L2TP
- Multiple MIP sessions



Important

Always refer to the Administration Guides for individual products for other possible session recovery and Interchassis Session Recovery (ICSR) support limitations.

When session recovery occurs, the system reconstructs the following subscriber information:

- Data and control state information required to maintain correct call behavior.
- A minimal set of subscriber data statistics; required to ensure that accounting information is maintained.
- A best-effort attempt to recover various timer values such as call duration, absolute time, and others.
- The idle time timer is reset to zero and the re-registration timer is reset to its maximum value for HA sessions to provide a more conservative approach to session recovery.

**Important**

Any partially connected calls (for example, a session where HA authentication was pending but has not yet been acknowledged by the AAA server) are not recovered when a failure occurs.

Additional ASR 5500 Hardware Requirements

Because session recovery requires numerous hardware resources, such as memory, control processors, NPU processing capacity, some additional hardware may be required to ensure that enough resources are available to fully support this feature.

**Important**

A minimum of four packet processing cards (three active and one standby) per individual chassis is required to use this feature.

To allow for complete session recovery in the event of a hardware failure during a packet processing card migration, a minimum of three active packet processing cards and two standby packet processing cards should be deployed.

To assist you in your network design and capacity planning, consider the following factors:

- Subscriber capacity is decreased depending on the hardware configuration. A fully configured chassis would experience a smaller decrease in subscriber capacity versus a minimally configured chassis.
- The amount by which control transaction processing capacity is reduced.
- The reduction in subscriber data throughput.
- The recovery time for a failed software task.
- The recovery time for a failed packet processing card.

A packet processing card migration may temporarily impact session recovery as hardware resources (memory, processors, etc.) that may be needed are not available during the migration. To avoid this condition, a minimum of two standby packet processing cards should be configured.

Configuring the System to Support Session Recovery

The following procedures allow you to configure the session recovery feature for either an operational system that is currently in-service (able to accept incoming calls) or a system that is out-of-service (not part of your production network and, therefore, not processing any live subscriber/customer data).

**Important**

The session recovery feature, even when the feature use key is present, is disabled by default on the system.

Enabling Session Recovery

As noted earlier, session recovery can be enabled on a system that is out-of-service (OOS) and does not yet have any contexts configured, or on an in-service system that is currently capable of processing calls. However, if the system is in-service, it must be restarted before the session recovery feature takes effect.

Enabling Session Recovery on an Out-of-Service System

The following procedure is for a system that does not have any contexts configured.

To enable the session recovery feature on an out-of-service system, follow the procedure below. This procedure assumes that you begin at the Exec mode prompt.

Step 1 At the Exec mode prompt, verify that the session recovery feature is enabled via the session and feature use licenses on the system by running the **show license info** command.

If the current status of the Session Recovery feature is Disabled, you cannot enable this feature until a license key is installed in the system.

Step 2 Use the following configuration example to enable session recovery.

```
configure
require session recovery
end
```

Note After you configure this command, you must save the configuration and then reload the chassis for the command to take effect. For information on saving the configuration file and reloading the chassis, refer to the System Administration Guide for your deployment.

Step 3 Save your configuration as described in *Verifying and Saving Your Configuration*.

The system, when started, enables session recovery, creates all mirrored "standby-mode" tasks, and performs packet processing card reservations and other operations automatically.

Step 4 After the system has been configured and placed in-service, you should verify the preparedness of the system to support this feature as described in [Viewing Session Recovery Status, on page 6](#)

Enabling Session Recovery on an In-Service System

When enabling session recovery on a system that already has a saved configuration, the session recovery commands are automatically placed before any service configuration commands in the configuration file.

To enable the session recovery feature on an in-service system, follow the procedure below. This procedure assumes that you begin at the Exec mode prompt.

Step 1 At the Exec mode prompt, verify that the session recovery feature is enabled via the session and feature use licenses on the system by running the **show license info** command:

If the current status of the Session Recovery feature is Disabled, You cannot enable this feature until a license key is installed in the system.

Step 2 Use the following configuration example to enable session recovery.

```
configure
  require session recovery
end
```

This feature does not take effect until after the system has been restarted.

Step 3 Save your configuration as described in *Verifying and Saving Your Configuration*.

Step 4 Perform a system restart by entering the **reload** command:

The following prompt appears:

```
Are you sure? [Yes|No]:
```

Confirm your desire to perform a system restart by entering **yes**.

The system, when restarted, enables session recovery and creates all mirrored "standby-mode" tasks, performs packet processing card reservations, and other operations automatically.

Step 5 After the system has been restarted, you should verify the preparedness of the system to support this feature as described in [Viewing Session Recovery Status, on page 6](#)

More advanced users may opt to simply insert the **require session recovery** command syntax into an existing configuration file using a text editor or other means, and then applying the configuration file manually. Exercise caution when doing this to ensure that this command is placed among the first few lines of any existing configuration file; it must appear before the creation of any non-local context.

Disabling the Session Recovery Feature

To disable the session recovery feature on a system, enter the **no require session recovery** command from the Global Configuration mode prompt.



Important If this command is issued on an in-service system, then the system must be restarted by issuing the **reload** command.

Viewing Session Recovery Status

To determine if the system is capable of performing session recovery, when enabled, enter the **show session recovery status verbose** command from the Exec mode prompt.

The output of this command should be similar to the examples shown below.

```
[local]host_name# show session recovery status
Session Recovery Status:
  Overall Status           : SESSMGR Not Ready For Recovery
  Last Status Update      : 1 second ago
```

```
[local]host_name# show session recovery status
Session Recovery Status:
  Overall Status           : Ready For Recovery
  Last Status Update      : 8 seconds ago
```

```
[local]host_name# show session recovery status verbose
```

Session Recovery Status:

```
Overall Status      : Ready For Recovery
Last Status Update : 2 seconds ago
```

```

-----sessmgr-----      -----aaamgr-----      demux
cpu state   active   standby   active   standby   active   status
-----
1/1 Active  2         1         1         1         0         Good
1/2 Active  1         1         0         0         0         Good
1/3 Active  1         1         3         1         0         Good
2/1 Active  1         1         1         1         0         Good
2/2 Active  1         1         0         0         0         Good
2/3 Active  2         1         3         1         0         Good
3/0 Active  0         0         0         0         1         Good (Demux)
3/2 Active  0         0         0         0         1         Good (Demux)
4/1 Standby 0         2         0         1         0         Good
4/2 Standby 0         1         0         0         0         Good
4/3 Standby 0         2         0         3         0         Good
[local]host_name#
```

Viewing Recovered Session Information

To view session state information and any session recovery status, enter the following command:

```
[local]host_name# show subscriber debug-info { callid id | msid id | username
name }
```

The following example shows the output of this command both before and after a session recovery operation has been performed. The "Redundancy Status" fields in this example have been bold-faced for clarity.

```
username: user1          callid: 01callb1          msid: 0000100003
Card/Cpu: 4/2
Sessmgr Instance: 7
Primary callline:
Redundancy Status: Original Session
  Checkpoints   Attempts   Success   Last-Attempt   Last-Success
  Full:         69         68         29800ms        29800ms
  Micro:        206        206        20100ms        20100ms
Current state: SMGR_STATE_CONNECTED
FSM Event trace:
  State          Event
  SMGR_STATE_OPEN      SMGR_EVT_NEWCALL
  SMGR_STATE_NEWCALL_ARRIVED  SMGR_EVT_ANSWER_CALL
  SMGR_STATE_NEWCALL_ANSWERED  SMGR_EVT_LINE_CONNECTED
  SMGR_STATE_LINE_CONNECTED    SMGR_EVT_LINK_CONTROL_UP
  SMGR_STATE_LINE_CONNECTED    SMGR_EVT_AUTH_REQ
  SMGR_STATE_LINE_CONNECTED    SMGR_EVT_IPADDR_ALLOC_SUCCESS
  SMGR_STATE_LINE_CONNECTED    SMGR_EVT_AUTH_SUCCESS
  SMGR_STATE_LINE_CONNECTED    SMGR_EVT_UPDATE_SESS_CONFIG
  SMGR_STATE_LINE_CONNECTED    SMGR_EVT_LOWER_LAYER_UP
Data Reorder statistics
Total timer expiry: 0          Total flush (tmr expiry): 0
  Total no buffers: 0          Total flush (no buffers): 0
  Total flush (queue full): 0  Total flush (out of range): 0
  Total flush (svc change): 0  Total out-of-seq pkt drop: 0
  Total out-of-seq arrived: 0
IPv4 Reassembly Statistics:
  Success: 0          In Progress: 0
  Failure (timeout): 0  Failure (no buffers): 0
  Failure (other reasons): 0
Redirected Session Entries: 2000  Allowed: 0
Current: 0
```

```

Added:                                0                                Deleted:
                                0
Revoked for use by different subscriber: 0
Peer callline:
Redundancy Status: Recovered Session
Checkpoints      Attempts      Success      Last-Attempt      Last-Success
Full:            0            0            0ms              0ms
Micro:          0            0            0ms              0ms
Current state: SMGR_STATE_CONNECTED
FSM Event trace:
State                                                    Event
SMGR_STATE_LINE_CONNECTED      SMGR_EVT_LOWER_LAYER_UP
SMGR_STATE_CONNECTED          SMGR_EVT_AUTH_REQ
SMGR_STATE_CONNECTED          SMGR_EVT_AUTH_SUCCESS
SMGR_STATE_CONNECTED          SMGR_EVT_REQ_SUB_SESSION
SMGR_STATE_CONNECTED          SMGR_EVT_RSP_SUB_SESSION
SMGR_STATE_CONNECTED          SMGR_EVT_ADD_SUB_SESSION
SMGR_STATE_CONNECTED          SMGR_EVT_AUTH_REQ
SMGR_STATE_CONNECTED          SMGR_EVT_AUTH_SUCCESS
SMGR_STATE_CONNECTED          SMGR_EVT_AUTH_REQ
SMGR_STATE_CONNECTED          SMGR_EVT_AUTH_SUCCESS
SMGR_STATE_CONNECTED          SMGR_EVT_AUTH_REQ
SMGR_STATE_CONNECTED          SMGR_EVT_AUTH_SUCCESS
SMGR_STATE_CONNECTED          SMGR_EVT_AUTH_REQ
SMGR_STATE_CONNECTED          SMGR_EVT_AUTH_SUCCESS
SMGR_STATE_CONNECTED          SMGR_EVT_AUTH_REQ
SMGR_STATE_CONNECTED          SMGR_EVT_AUTH_SUCCESS
Data Reorder statistics
Total timer expiry:            0            Total flush (tmr expiry): 0
Total no buffers:              0            Total flush (no buffers): 0
Total flush (queue full):      0            Total flush (out of range):0
Total flush (svc change):      0            Total out-of-seq pkt drop: 0
Total out-of-seq arrived:      0
IPv4 Reassembly Statistics:
Success:                        0            In Progress:              0
Failure (timeout):              0            Failure (no buffers):     0
Failure (other reasons):        0
Redirected Session Entries:
Allowed:                        2000         Current:                   0
Added:                          0            Deleted:                   0
Revoked for use by different subscriber: 0

```

Recovery Control Task Statistics

Recovery Control Task (RCT) statistics show the following:

- Recovery action taken – Migration, Shutdown, Switchover
- Type of event – Planned or Unplanned
- From card to card – slot numbers
- Start time – YYYY-MMM-DD+hh:mm:sss.sss
- Duration – seconds
- Card failure device (such as CPU n)
- Card failure reason
- Card is in usable state or not failed
- Recovery action status – Success or failure reason
- If recovery action failed, failure time stamp
- If recovery action failed, failure task facility name

- If recovery action failed, failure instance number

show rct stats Command

The Exec mode **show rct stats** command employs the following syntax:

```
[local]host_name# show rct stats [verbose]
```

Without the **verbose** keyword, a summary output is displayed as show in the example below:

RCT stats details (Last 1 Actions)

#	Action	Type	From	To	Start Time	Duration	Status
1	Migration(st)	Planned	2	1	2016-Jul-12+13:12:21.865	0.003 sec	Success

RCT stats summary

```
-----
Migrations = 0
Management Card: 0 Average time: 0.000 sec
Packet Card : 1 Average time: 0.006 sec
Switchovers = 1, Average time - 25.855 sec
```

With the **verbose** keyword the detailed statistics show in [Sample Output for show rct stats verbose](#), on page 9 are provided.

Sample Output for show rct stats verbose

```
[local]host_name# show rct stats verbose
```

RCT stats Details (Last 5 Actions)

Stats 1:

```
Action      : Migration
Type        : Planned
From        : 5
To          : 6
Start Time   : 2017-Apr-04+03:02:00.132
Failure Reason : CPU_CRITICAL_TASK_FAILURE
Failure Device : CPU_0
Is Card Usable : Yes
Recovery Status : Success
Facility     : N.A
Instance    : N.A
Duration     : 066.050 sec
Graceful     : Enabled
Recovered [1] : [f:sessmgr, i:6, cpu:50, pid:13170]
Recovered [2] : [f:sessmgr, i:3, cpu:50, pid:13167]
```

RCT stats Details (Last 5 Actions)

Stats 2:

```
Action      : Shutdown
From        : 12
To          : 13
Start Time   : 2017-Apr-04+03:02:10.100
Is Card Usable : Yes
Failure Reason : NPU_LC_CONNECT_TOP_FAIL
Failure Device : PAC_LC_CONNECT_HARDWARE
Recovery Status : Success
```

Sample Output for show rct stats verbose

```

Facility      : N.A
Instance     : N.A
Duration     : 002.901 sec
Graceful     : Enabled
Recovered [1]: [f:sessmgr, i:6, cpu:50, pid:13170
Recovered [2]: [f:sessmgr, i:3, cpu:50, pid:13167]

Stats 3:
Action       : Migration
From         : 7
To           : 11
Start Time   : 2017-Apr-04+03:03:40.120
Is Card Usable : Yes
Failure Reason : N.A.
Failure Device : N.A
Recovery Status : Success
Facility     : N.A
Instance     : N.A
Duration     : 003.423 sec
Graceful     : Enabled
Recovered [1]: [f:sessmgr, i:6, cpu:50, pid:13170
Recovered [2]: [f:sessmgr, i:3, cpu:50, pid:13167]

Stats 4:
Action       : Migration
From         : 7
To           : 11
Start Time   : 2017-Apr-04+03:03:41.256
Is Card Usable : Yes
Failure Reason : N.A.
Failure Device : N.A
Recovery Status : TASK_MIGRATION_FAIL_PREMIGRATE
Facility     : vpnmgr
Instance     : 13
Duration     : 005.222 sec
Graceful     : Enabled
Recovered [1]: [f:sessmgr, i:6, cpu:50, pid:13170
Recovered [2]: [f:sessmgr, i:3, cpu:50, pid:13167]

Stats 5:
Action       : Migration
From         : 6
To           : 7
Start Time   : 2017-Apr-04+04:18:30.106
Is Card Usable : Yes
Failure Reason : N.A.
Failure Device : N.A
Recovery Status : TASK_MIGRATION_FAIL_RENAME
Facility     : sessmgr
Instance     : 63
Duration     : 004.134 sec
Graceful     : Enabled
Recovered [1]: [f:sessmgr, i:6, cpu:50, pid:13170
Recovered [2]: [f:sessmgr, i:3, cpu:50, pid:13167]

RCT stats Summary
-----
Migrations = 3, Average time = 4.260 sec
Switchovers = 0

```