

ASR5x00会话管理器任务 — 功能、故障、恢复操作和故障日志的说明

目录

[简介](#)

[软件架构：专为恢复能力而设计](#)

[什么是崩溃？](#)

[会话管理器崩溃的影响](#)

[操作员何时应该感到担忧？](#)

[如何知道是否发生了崩溃？](#)

[故障记录架构](#)

[管理卡之间崩溃事件和Minicore的同步](#)

[命令](#)

[摘要](#)

简介

本文档介绍并解释思科聚合服务路由器(ASR)5x00系列的软件可靠性、服务可用性和故障切换功能。它介绍了ASR5x00上软件崩溃的定义以及软件崩溃的影响。文章接着指出，即使发生意外软件崩溃，ASR5x00如何凭借固有的软件恢复能力和可用性功能实现“运营级”可用性目标。移动用户不必考虑服务的可用性。思科的目标是避免因任何单个硬件或软件故障而丢失会话，包括丢失完整的系统，换句话说，即语音等级可靠性。ASR5x00上的软件可靠性功能旨在实现“运营级”服务可用性的目标，即使在运营商网络中可能发生不可预见的故障时也是如此。

软件架构：专为恢复能力而设计

ASR5x00具有分布在数据包服务卡(PSC)或数据处理卡(DPC)和系统管理卡(SMC)或管理和I/O(MIO)卡上的软件任务集，这些软件任务旨在执行各种特定功能。

例如，会话管理器任务负责处理一组用户的会话，并对用户流量执行内联服务，如点对点(P2P)、深度数据包检测(DPI)等。身份验证、授权和记帐(AAA)管理器任务负责生成计费事件，以记录用户流量使用情况等。会话管理器和AAA管理器任务在PSC/DPC卡上运行。

SMC/MIO卡保留用于操作和维护(O&M)和平台相关任务。ASR5x00系统被虚拟地划分为不同的软件子系统，例如用于处理用户会话的会话子系统和负责IP地址分配、路由等的VPN子系统。每个子系统都有一个控制器任务，该任务负责监控其控制的子系统的运行状况。控制器任务在SMC/MIO卡上运行。会话管理器和AAA管理器任务配对，以处理用户的会话，以便进行控制、数据流量和计费。当系统中启用会话恢复时，每个会话管理器任务都会备份其用户状态集的状态，并备份一个对等AAA管理器任务，以在会话管理器崩溃时恢复。

什么是崩溃？

如果ASR5x00中的任务在正常运行期间遇到故障情况，则可能会崩溃。ASR5x00中的崩溃或软件故障定义为系统中任务的意外退出或终止。如果软件代码尝试访问禁止的内存区域（如损坏的数据结构），遇到代码中不预期的情况（如无效状态转换）等，则会发生崩溃。如果任务对系统监控任务无响应且监控器尝试终止和重新启动该任务，也可触发崩溃。当通过CLI命令或系统监控器强制转储任务的当前状态以分析任务状态时，也可以在系统中显式触发崩溃事件（而不是意外）。当系统控制器任务重新启动时，也可能发生预期的崩溃事件，以便可能纠正管理器任务反复失败的情况。

会话管理器崩溃的影响

在正常操作下，会话管理器任务处理会话的一组用户会话和关联数据流量，以及处理这些用户会话的计费的AAA管理器任务。当会话管理器发生崩溃时，会话管理器将不再存在于系统中。如果在系统中启用了会话恢复，则会使备用会话管理器任务在同一PSC/DPC卡中变为活动状态。此新会话管理器任务在与对等AAA管理器任务通信时重新声明用户会话。恢复操作范围从50毫秒到几秒，具体取决于崩溃时会话管理器中处于活动状态的会话数以及卡上的整体CPU负载等。在此操作中，在原始会话管理器中已建立的用户会话中没有丢失。崩溃时正在建立的任何用户会话都可能由于协议重新传输等原因而恢复。在崩溃时通过系统转换的任何数据包都可以被假定与网络连接的通信实体的网络丢失相关联并且将被重新传输并且连接将由新的会话管理器进行。会话管理器承载的会话的计费信息将保留在对等AAA管理器中。

操作员何时应该感到担忧？

当会话管理器崩溃时，恢复过程会如前所述发生，系统的其余部分不会受此事件影响。一个会话管理器中的崩溃不会影响其他会话管理器。如果同一PSC/DPC卡上的多个会话管理器任务同时崩溃或在10分钟内相互崩溃，则可能会丢失会话，因为系统可能无法以足够快的速度启动新会话管理器来替代崩溃的任务。这对应于会话丢失的双故障场景。当恢复不可行时，会话管理器只需重新启动即可接受新会话。

当给定会话管理器反复崩溃（例如，它反复遇到相同的故障情况）时，会话控制器任务会记录并重新启动自己以尝试恢复子系统。如果会话控制器任务无法稳定会话子系统，并在此工作中继续重新启动，升级的下一步是系统切换到备用SMC/MIO卡。如果出现不存在备用SMC/MIO卡的不可能事件，或者如果在切换操作中遇到故障，系统会自行重新启动。

会话管理器还维护每个接入点名称(APN)、服务、功能等的统计信息，这些统计信息在发生崩溃时将永久丢失。因此，定期收集批量统计数据的外部实体将在发生一个或多个崩溃时观察统计数据的下滑。这可以表现为在时间轴上绘制的统计数据的图形表示中的凹陷。

注意：使用7-14 PSC或4-10 DPC卡填充的典型机箱具有大约120-160个会话管理器，具体取决于PSC/DPC卡的数量，而单次崩溃将导致大约1/40或1/80的统计数据。当备用会话管理器接管时，它开始再次从零开始累计统计信息。

如何知道是否发生了崩溃？

崩溃将触发SNMP陷阱事件到网络监控站，如事件监控服务(EMS)和系统日志事件。使用show crash list命令也可以观察系统中发生的崩溃。请注意，此命令会列出意外和预期的崩溃事件，如前

所述。这两种崩溃事件可以通过描述每次崩溃的报头进行区分。

以下日志消息表示任务崩溃后会成功恢复会话：

```
"Death notification of task <name>/<instance id> on <card#>/<cpu#> sent to parent
task <parent name>/<instance id> with failover of <task name>/<instance id>
on <card#>/<cpu#>"
```

此日志消息指示无法恢复的任务崩溃：

```
"Death notification of task <name>/<instance id> on <card#>/<cpu#> sent to parent
task <parent name>/<instance id>"
```

总之，启用会话恢复后，大多数情况下不会注意到崩溃，因为它们对用户没有影响。必须输入CLI命令，或查看日志或SNMP通知，以检测崩溃的发生。

例如：

```
***** show crash list *****
Tuesday May 26 05:54:14 BDT 2015
=== =====
# Time Process Card/CPU/ SW HW_SER_NUM
PID VERSION MIO / Crash Card
=== =====

1 2015-May-07+11:49:25 sessmgr 04/0/09564 17.2.1 SAD171600WS/SAD172200MH
2 2015-May-13+17:40:16 sessmgr 09/1/05832 17.2.1 SAD171600WS/SAD173300G1
3 2015-May-23+09:06:48 sessmgr 03/1/31883 17.2.1 SAD171600WS/SAD1709009P
4 2015-May-25+15:58:59 sessmgr 09/1/16963 17.2.1 SAD171600WS/SAD173300G1
5 2015-May-26+01:15:15 sessmgr 04/0/09296 17.2.1 SAD171600WS/SAD172200MH

***** show snmp trap history verbose *****
Fri May 22 19:43:10 2015 Internal trap notification 1099 (ManagerRestart) facility
sessmgr instance 204 card 9 cpu 1
Fri May 22 19:43:29 2015 Internal trap notification 73 (ManagerFailure) facility
sessmgr instance 204 card 9 cpu 1
Fri May 22 19:43:29 2015 Internal trap notification 150 (TaskFailed) facility
sessmgr instance 204 on card 9 cpu 1
Fri May 22 19:43:29 2015 Internal trap notification 151 (TaskRestart) facility
sessmgr instance 204 on card 9 cpu 1
Fri May 22 19:43:30 2015 Internal trap notification 183 (SessMgrRecoveryComplete)
Slot Number 9 Cpu Number 1 fetched from aaa mgr 1755 prior to audit 1755 passed
audit 1754 calls recovered 1754 all call lines 1754 time elapsed ms 1108.
Fri May 22 19:43:32 2015 Internal trap notification 1099 (ManagerRestart) facility
sessmgr instance 204 card 9 cpu 1
Fri May 22 19:44:49 2015 Internal trap notification 73 (ManagerFailure) facility
sessmgr instance 236 card 7 cpu 0
Fri May 22 19:44:49 2015 Internal trap notification 150 (TaskFailed) facility
sessmgr instance 236 on card 7 cpu 0
Fri May 22 19:44:49 2015 Internal trap notification 151 (TaskRestart) facility
sessmgr instance 236 on card 7 cpu 0
Fri May 22 19:44:51 2015 Internal trap notification 183 (SessMgrRecoveryComplete)
Slot Number 7 Cpu Number 0 fetched from aaa mgr 1741 prior to audit 1741 passed audit
1737 calls recovered 1737 all call lines 1737 time elapsed ms 1047.
Fri May 22 19:44:53 2015 Internal trap notification 1099 (ManagerRestart) facility
sessmgr instance 236 card 7 cpu 0
Fri May 22 19:50:04 2015 Internal trap notification 73 (ManagerFailure) facility
sessmgr instance 221 card 2 cpu 1
: Fri May 22 19:50:04 2015 Internal trap notification 150 (TaskFailed) facility
```

```
sessmgr instance 221 on card 2 cpu 1
Fri May 22 19:50:04 2015 Internal trap notification 151 (TaskRestart) facility
sessmgr instance 221 on card 2 cpu 1
Fri May 22 19:50:05 2015 Internal trap notification 183 (SessMgrRecoveryComplete)
Slot Number 2 Cpu Number 1 fetched from aaa mgr 1755 prior to audit 1755 passed
audit 1749 calls recovered 1750 all call lines 1750 time elapsed ms 1036.
```

```
***** show snmp trap history verbose *****
```

```
Fri May 22 19:43:10 2015 Internal trap notification 1099 (ManagerRestart) facility
sessmgr instance 204 card 9 cpu 1
Fri May 22 19:43:29 2015 Internal trap notification 73 (ManagerFailure) facility
sessmgr instance 204 card 9 cpu 1
Fri May 22 19:43:29 2015 Internal trap notification 150 (TaskFailed) facility
sessmgr instance 204 on card 9 cpu 1
Fri May 22 19:43:29 2015 Internal trap notification 151 (TaskRestart) facility
sessmgr instance 204 on card 9 cpu 1
Fri May 22 19:43:30 2015 Internal trap notification 183 (SessMgrRecoveryComplete)
Slot Number 9 Cpu Number 1 fetched from aaa mgr 1755 prior to audit 1755 passed
audit 1754 calls recovered 1754 all call lines 1754 time elapsed ms 1108.
Fri May 22 19:43:32 2015 Internal trap notification 1099 (ManagerRestart) facility
sessmgr instance 204 card 9 cpu 1
Fri May 22 19:44:49 2015 Internal trap notification 73 (ManagerFailure) facility
sessmgr instance 236 card 7 cpu 0
Fri May 22 19:44:49 2015 Internal trap notification 150 (TaskFailed) facility
sessmgr instance 236 on card 7 cpu 0
Fri May 22 19:44:49 2015 Internal trap notification 151 (TaskRestart) facility
sessmgr instance 236 on card 7 cpu 0
Fri May 22 19:44:51 2015 Internal trap notification 183 (SessMgrRecoveryComplete)
Slot Number 7 Cpu Number 0 fetched from aaa mgr 1741 prior to audit 1741 passed
audit 1737 calls recovered 1737 all call lines 1737 time elapsed ms 1047.
Fri May 22 19:44:53 2015 Internal trap notification 1099 (ManagerRestart) facility
sessmgr instance 236 card 7 cpu 0
Fri May 22 19:50:04 2015 Internal trap notification 73 (ManagerFailure) facility
sessmgr instance 221 card 2 cpu 1
: Fri May 22 19:50:04 2015 Internal trap notification 150 (TaskFailed) facility
sessmgr instance 221 on card 2 cpu 1
Fri May 22 19:50:04 2015 Internal trap notification 151 (TaskRestart) facility
sessmgr instance 221 on card 2 cpu 1
Fri May 22 19:50:05 2015 Internal trap notification 183 (SessMgrRecoveryComplete)
) Slot Number 2 Cpu Number 1 fetched from aaa mgr 1755 prior to audit 1755 passed
audit 1749 calls recovered 1750 all call lines 1750 time elapsed ms 1036.
```

```
***** show logs *****
```

```
2015-May-25+23:15:53.123 [sitmain 4022 info] [3/1/4850 <sitmain:31> sittask.c:4762]
[software internal system critical-info syslog] Readdress requested for facility
sessmgr instance 5635 to instance 114
2015-May-25+23:15:53.122 [sitmain 4027 critical] [3/1/4850 <sitmain:31>
crash_mini.c:908] [software internal system callhome-crash] Process Crash Info:
time 2015-May-25+17:15:52(hex time 556358c8) card 03 cpu 01 pid 27118 procname
sessmgr crash_details
Assertion failure at acs/acsmgr/analyzer/ip/acs_ip_reasm.c:2970
Function: acsmgr_deallocate_ipv4_frag_chain_entry()
Expression: status == SN_STATUS_SUCCESS
Procllet: sessmgr (f=87000,i=114)
Process: card=3 cpu=1 arch=X pid=27118 cpu=~17% argv0=sessmgr
Crash time: 2015-May-25+17:15:52 UTC
Recent errno: 11 Resource temporarily unavailable
Stack (11032@0xfffffb000):
[ffffe430/X] __kernel_vsyscall() sp=0xfffffb28
[0af1delf/X] sn_assert() sp=0xfffffb68
[0891e137/X] acsmgr_deallocate_ipv4_frag_chain_entry() sp=0xffffbde8
[08952314/X] acsmgr_ip_frag_chain_destroy() sp=0xffffbee8
[089d87d1/X] acsmgr_process_tcp_packet() sp=0xffffc568
```

```
[089da270/X] acs_process_tcp_packet_normal_path() sp=0xffffc5b8
[089da3fd/X] acs_tcp_analyzer() sp=0xffffc638
[0892fb39/X] do_acsmgr_process_packet() sp=0xffffc668
[08940045/X] acs_ip_lean_path() sp=0xffffc6b8
[0887e309/X] acsmgr_data_receive_merge_mode() sp=0xffffc9d8
[0887f323/X] acs_handle_datapath_events_from_sm_interface() sp=0xffffca08
[037c2e1b/X] sessmgr_sef_initiate_data_packet_ind() sp=0xffffca88
[037c2f50/X] sessmgr_pcc_intf_send_data_packet_ind() sp=0xffffcaf8
[061de74a/X] sessmgr_pcc_fwd_packet() sp=0xffffcb58
[0627c6a4/X] sessmgr_ipv4_process_inet_pkt_part2_slow() sp=0xffffcf68
[06318343/X] sessmgr_ipv4_process_inet_pkt_pgw_ggsn() sp=0xffffd378
[0632196c/X] sessmgr_med_ipv4_data_received() sp=0xffffd418
[0633da9a/X] sessmgr_med_data_receive() sp=0xffffd598
[0afb977c/X] sn_epoll_run_events() sp=0xffffd5e8
[0afbdeb8/X] sn_loop_run() sp=0xffffda98
[0ad2b82d/X] main() sp=0xffffdb08
```

```
2015-May-25+23:15:53.067 [rct 13038 info] [5/0/7174 <rct:0> rct_task.c:305]
[software internal system critical-info syslog] Death notification of task
sessmgr/114 on 3/1 sent to parent task sessctrl/0 with failover of sessmgr/5635 on 3/1
2015-May-25+23:15:53.065 [evlog 2136 info] [5/0/7170 <evlogd:0> odule_persist.c:3102]
[software internal system critical-info syslog] Evlogd crashlog: Request received to
check the state of persistent crashlog.
2015-May-25+23:15:53.064 [sitmain 4099 info] [3/1/4850 <sitmain:31> crash_mini.c:765]
[software internal system critical-info syslog] have mini core, get evlogd status for
logging crash file 'crashdump-27118'
2015-May-25+23:15:53.064 [sitmain 4017 critical] [3/1/4850 <sitmain:31> sitproc.c:1544]
[software internal system syslog] Process sessmgr pid 27118 died on card 3 cpu 1
signal=6 wstatus=0x86
2015-May-25+23:15:53.048 [sitmain 4074 trace] [5/0/7168 <sitparent:50> crashd.c:1130]
[software internal system critical-info syslog] Crash handler file transfer starting
(type=2 size=0 child_ct=1 core_ct=1 pid=23021)
2015-May-25+23:15:53.047 [system 1001 error] [6/0/9727 <evlogd:1> evlgd_syslogd.c:221]
[software internal system syslog] CPU[3/1]: xmitcore[21648]: Core file transmitted to
card 5 size=663207936 elapsed=0sec:908ms
2015-May-25+23:15:53.047 [system 1001 error] [5/0/7170 <evlogd:0> evlgd_syslogd.c:221]
[software internal system syslog] CPU[3/1]: xmitcore[21648]: Core file transmitted to
card 5 size=663207936 elapsed=0sec:908ms
2015-May-25+23:15:53.047 [sitmain 4080 info] [5/0/7168 <sitparent:50> crashd.c:1091]
[software internal system critical-info syslog] Core file transfer to SPC complete,
received 8363207936/0 bytes
```

```
***** show session recovery status verbose *****
Tuesday May 26 05:55:26 BDT 2015
Session Recovery Status:
Overall Status : Ready For Recovery
Last Status Update : 8 seconds ago
```

```
----sessmgr--- ----aaamgr---- demux
cpu state active standby active standby active status
-----
1/0 Active 24 1 24 1 0 Good
1/1 Active 24 1 24 1 0 Good
2/0 Active 24 1 24 1 0 Good
2/1 Active 24 1 24 1 0 Good
3/0 Active 24 1 24 1 0 Good
3/1 Active 24 1 24 1 0 Good
4/0 Active 24 1 24 1 0 Good
4/1 Active 24 1 24 1 0 Good
5/0 Active 0 0 0 0 14 Good (Demux)
7/0 Active 24 1 24 1 0 Good
7/1 Active 24 1 24 1 0 Good
8/0 Active 24 1 24 1 0 Good
```

```
8/1 Active 24 1 24 1 0 Good
9/0 Active 24 1 24 1 0 Good
9/1 Active 24 1 24 1 0 Good
10/0 Standby 0 24 0 24 0 Good
10/1 Standby 0 24 0 24 0 Good
```

故障记录架构

故障日志记录与软件故障（完整核心转储）相关的所有可能信息。由于其大小，无法将其存储在系统内存中。因此，仅当系统配置了指向本地设备或网络服务器的URL时，才会生成这些日志。

故障日志是故障事件信息的持久存储库。每个事件都进行编号并包含与CPU（小型）、网络处理单元(NPU)或内核崩溃相关的文本。记录的事件记录到固定长度的记录中，并存储在 /flash/crashlog2 中。

每当发生崩溃时，都会存储此崩溃信息：

1. 事件记录存储在 /flash/crashlog2 文件（故障日志）中。
2. 关联的 minicore、NPU 或内核转储文件存储在 /flash/crsh2 目录中。
3. 完整核心转储存储在用户配置的目录中。

管理卡之间崩溃事件和Minicore的同步

崩溃日志是每个管理卡的唯一，因此，如果卡“8”处于活动状态时发生崩溃，则会登录卡“8”。随后的切换将不再显示日志中的崩溃。要检索此故障，必须将交换机切换回卡“8”。故障事件日志和转储对活动和备用管理卡是唯一的，因此，如果活动卡发生故障，则故障事件日志和相关转储将仅存储在活动卡上。备用卡上不提供此故障信息。每当由于活动卡崩溃而卡切换，并且故障信息不再显示在接管的卡上时，故障信息只能从当前活动卡中检索。要检索另一卡的故障列表，需要再次进行切换。为避免此切换并从备用卡获取故障信息，需要在两个管理卡之间同步并维护最新的故障信息。

到达的崩溃事件将发送到备用SMC/MIO，并以类似方式保存到备用的崩溃日志文件中。需要使用 rsync 命令将活动SMC/MIO闪存上的Minicore、NPU或内核转储同步到备用SMC/MMIO。当通过 CLI 命令删除 crashlog 条目或整个列表时，应在主用和备用SMC/MIO上清除该条目。对内存没有影响。所有与崩溃相关的同步活动都将由备用SMC/MIO卡的事件完成，因为备用事件负载较少且备用卡有足够的空间进行同步活动。因此，系统的性能不会受到影响。

命令

以下命令可用于排除故障：

```
#show support details
```

```
#show crash list
```

```
#show logs
```

```
#show snmp trap history verbose
```

```
#show session recovery status verbose
```

```
#show task resources facility sessmgr instance <>
```

```
#show task resources facility sessmgr all
```

崩溃后生成核心文件。通常，操作员将其存储在外部服务器中。核心文件名通常类似于crash-
<Cardnum>-<CPU Num>-<Hex timestamp>-coree.gcrash-09-00-5593a1b8-core。

每当发生崩溃时，都会存储此崩溃信息：

- 事件记录存储在/flash/crashlog2文件（故障日志）中。
- 关联的minicore、NPU或内核转储文件存储在/flash/crsh2目录中。

摘要

所有ASR5x00软件都设计用于处理预见的情况/事件和不可预见的情况/事件。在思科努力拥有完美软件的同时，难免会出现错误，并可能发生崩溃。这就是会话恢复功能如此重要的原因。思科力求完美，可最大限度地减少崩溃的发生，会话恢复可让会话在崩溃后继续。尽管如此，思科仍必须继续努力实现完美的软件。减少崩溃将降低同时发生多次崩溃的可能性。虽然会话恢复可以无缝地消除单次崩溃，但从多个同时崩溃中恢复的设计有些不同。操作员很少（或从不）遇到多次同时崩溃，但如果发生这种情况，ASR5x00旨在将系统完整性恢复为最高优先级，可能会牺牲一些用户会话。

。