

# Менеджер сеанса ASR5x00 задачи - описание функции, катастрофического отказа, операций восстановления и крешлогов

## Содержание

[Введение](#)

[Архитектура программного обеспечения: Разработанный для упругости](#)

[Что такое катастрофический отказ?](#)

[Эффекты катастрофического отказа менеджера сеанса](#)

[Когда должен быть заинтересован оператор?](#)

[Как знать, произошел ли катастрофический отказ?](#)

[Архитектура Регистрации катастрофического отказа](#)

[Синхронизация событий катастрофического отказа и миниядер между картами управления](#)

[Команды](#)

[Сводка](#)

## Введение

Этот документ описывает и объясняет надежность ПО, доступность службы и функции аварийного переключения Маршрутизатора агрегации (ASR) Cisco 5x00 Серия. Это представляет определение для сбоя в программном обеспечении на ASR5x00 и эффектах сбоя в программном обеспечении. Статья продолжает устанавливать, что даже в случае неожиданных сбоев в программном обеспечении, как ASR5x00 в состоянии забить гол доступности в связи с "класса носителя" к свойственной упругости программного обеспечения и функциям доступности. Мобильному абоненту никогда не придется думать о доступности сервиса. Целью Cisco не является никакая потеря сеанса ни из-за каких одиночных аппаратных средств или сбоев программного обеспечения, которые включают потерю полной системы, другими словами - надежность полосы пропускания для передачи голоса. Функции надежности ПО на ASR5x00 предназначены, чтобы быть в состоянии достигнуть целей для доступности службы "класса носителя" даже в случаях, где непредвиденные сбои могли бы произойти в сети оператора.

## Архитектура программного обеспечения: Разработанный для упругости

ASR5x00 имеет набор программных задач, распределенных через Карту служб пакетной передачи (PSC) или Карту обработки данных (DPC) и Карту управления системой (SMC) или менеджмент и ввод-вывод (MIO) карты, которые разработаны для выполнения множества определенных функций.

Например, задача менеджера сеанса ответственна за обработку сеансов для ряда абонентов и выполнять встроенные сервисы такой как Одноранговые (P2P), Глубокая проверка пакетов (DPI), и так далее, на трафике пользователя. Задача менеджера Аутентификации, авторизации и учета (AAA) ответственна за генерацию составления счетов событий для записи использования трафика абонента и так далее. Менеджер сеанса и менеджер AAA задачи работают на карте PSC/DPC.

Карта SMC/MIO зарезервирована для Эксплуатации и обслуживания (O&M), и платформа отнеслась задачи. Система ASR5x00 фактически разделена в другие программные подсистемы, такие как подсистема сеанса для обработки сеансов абонента и подсистемы VPN, которая ответственна за присвоение IP-адреса, маршрутизацию, и так далее. Каждая подсистема имеет задачу контроллера, которая наблюдает за состоянием подсистемы, которой это управляет. Задачи контроллера работают на карте SMC/MIO. Менеджер сеанса и менеджер AAA, задачи соединены вместе для обработки сеанса абонента для контроля, трафика данных и выставлений счетов. Когда восстановление сеанса включено в системе, каждая задача менеджера сеанса выполняет резервное копирование состояние своего набора состояний абонента с одноранговым менеджером AAA задача, которая будет восстановлена в случае катастрофического отказа менеджера сеанса.

## Что такое катастрофический отказ?

Задача в ASR5x00 может потенциально завершиться катастрофическим отказом, если это встречается с условием отказа во время нормальной работы. Катастрофический отказ или программный сбой в ASR5x00 определены, чтобы быть *неожиданным* выходом или завершением задачи в системе. Катастрофический отказ может произойти, если программный код пытается получить доступ к памяти области, которые запрещены (такие как поврежденные структуры данных), встречается с условием в коде, который не ожидается (такие как недопустимое изменение состояния) и так далее. Катастрофический отказ может также быть инициирован, если задача становится безразличной к задаче системного монитора, и монитор пытается уничтожить и перезапустить задачу. Событие катастрофического отказа может также быть явно инициировано (в противоположность неожиданному) в системе, когда задача вынуждена формировать дамп своего текущего состояния командой CLI или системным монитором для анализа состояния задачи. Ожидаемый случай катастрофического отказа может также произойти, когда задачи системного контроллера перезапускают себя для потенциального исправления ситуации с задачей менеджера, которая неоднократно отказывает.

## Эффекты катастрофического отказа менеджера сеанса

Под нормальной работой задача менеджера сеанса обрабатывает ряд сеансов абонента и привязанного трафика данных для сеансов наряду с пиринговым менеджером AAA задача, которая обрабатывает составление счетов за те сеансы абонента. Когда катастрофический отказ менеджера сеанса происходит, он прекращает существование в системе. Если восстановление сеанса включено в системе, резервная задача менеджера сеанса сделана стать активной в той же карте PSC/DPC. Эта новая задача менеджера сеанса восстанавливает сеансы абонента, поскольку она передает с одноранговым менеджером AAA задачу. Операция восстановления колеблется от 50 мс до нескольких секунд, зависящих от количества сеансов, которые были активны в менеджере сеанса во время катастрофического отказа и полной Загрузки ЦПУ на карте и так далее. Нет никакой потери

на сеансах абонента, которые были уже установлены в менеджере исходного сеанса в этой операции. Любой сеанс абонента, который был в процессе установления во время катастрофического отказа, будет, вероятно, также восстановлен из-за повторных передач протокола и так далее. Любые пакеты данных, которые были в переходе через систему во время катастрофического отказа, могут предполагаться быть привязанными к сети loss взаимодействующими объектами сетевого подключения и будут ретранслироваться, и соединение будет продолжено новым менеджером сеанса. Информация о выставлении счетов для сеансов, которые несет менеджер сеанса, будет сохранена в одноранговом менеджере AAA.

## Когда должен быть заинтересован оператор?

Когда катастрофический отказ менеджера сеанса происходит, процедура восстановления происходит, как описано ранее, и остаток системы остается незатронутым этим событием. Катастрофический отказ в одном менеджере сеанса не влияет на других менеджеров сеанса. Как руководство оператору, если менеджер несколько сеансов определяет задачу *на том же сбое карты PSC/DPC* одновременно или в течение 10 минут друг после друга, могло бы быть потерей сеансов, поскольку система не могла бы быть в состоянии запустить новых менеджеров сеанса достаточно быстро для занимания места завершившихся катастрофическим отказом задач. Это соответствует сценарию двойного отказа, где может произойти потеря сеансов. Когда восстановление не выполнимо, менеджер сеанса просто перезапущен и готов принять новые сеансы.

То, когда данный менеджер сеанса неоднократно завершается катастрофическим отказом (такие как он встречается с тем же условием отказа много раз), задача контроллера сессий обращает внимание и перезапускает себя в попытке восстановить подсистему. Если задача контроллера сессий неспособна стабилизировать подсистему сеанса и перезапускает себя постоянно в этом усилии, следующий шаг в эскалации для системы для переключений в резервную карту SMC/MIO. Если со сбоем встречаются в операции переключателя, системные перезагрузки самой, в маловероятном случае, что нет никакой резервной карты SMC/MIO или.

Менеджеры сеанса также поддерживают статистику для Названия каждой точки доступа (APN), Сервисы, *functionalites*, и так далее который будет постоянно потерян, когда произойдет катастрофический отказ. Поэтому внешний объект, который собирает *bulkstats* периодически, будет наблюдать падение в статистике, когда произойдут один или несколько сбоев. Это может проявить как падение в графическом представлении статистики, дистиллируемой ось времени.

**Примечание:** Типичное шасси, заполненное с 7-14 пк или 4-10 карт DPC, имеют приблизительно 120-160 менеджеров сеанса, зависящих от количества карт PSC/DPC, и одиночный катастрофический отказ приведет к потере приблизительно  $\frac{1}{40}$ -го или  $\frac{1}{80}$ -го из статистики. Когда резервный менеджер сеанса вступает во владение, это начинает накапливать статистику снова от нуля.

## Как знать, произошел ли катастрофический отказ?

Катастрофический отказ инициирует событие trap-сообщения SNMP к станции мониторинга

сети, такой как Сервис мониторинга событий (EMS) и событиями системного журнала (syslog). Сбои, которые произошли в системе, могут также наблюдаться с командой **списка катастрофического отказа показа**. Обратите внимание на то, что это списки команд и неожиданные и ожидаемые события катастрофического отказа, как описано ранее. Эти два события типов аварии можно отличить посредством заголовка, который описывает каждый катастрофический отказ.

Катастрофический отказ задачи, придерживавшийся восстановлением успешного сеанса, обозначен этим сообщением журнала:

```
"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#>"
```

Катастрофический отказ задачи, который не мог восстановиться, обозначен этим сообщением журнала:

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

Таким образом, с включенным восстановлением сеанса, в большинстве случаев сбои не будут замечены, потому что они не оказывают влияния абонента. Нужно ввести команду CLI или посмотреть на журналы или уведомление SNMP для обнаружения любого возникновения сбоев.

Пример:

```
***** 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@0xffffb000):
[ffffe430/X] __kernel_vsyscall() sp=0xffffbd28
```

```
[0af1de1f/X] sn_assert() sp=0xffffbd68
[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
```

## Архитектура Регистрации катастрофического отказа

Крешлоги делают запись всей возможной информации, которые принадлежат сбою в программном обеспечении (полный дамп основной памяти). Из-за их размера, они не могут быть сохранены в системной памяти. Поэтому эти журналы только генерируются, если система настроена с URL, который указывает к локальному устройству или серверу сети, где может быть сохранен журнал.

Крешлог является персистентным репозиторием сведений о событии катастрофического отказа. Каждое событие пронумеровано и содержит текст, привязанный к ЦП (миниядро), модуль сетевой обработки (NPU) или катастрофический отказ ядра. Зарегистрированные события зарегистрированы в записи фиксированной длины и сохранены в `/flash/crashlog2`.

Каждый раз, когда катастрофический отказ происходит, эти сведения об аварийном отказе сохраняются:

1. Запись события сохранена в `/flash/crashlog2` файле (крешлог).
2. Связанное миниядро, NPU или файл дампа ядра сохранены в `/flash/crsh2` каталоге.
3. Полный дамп основной памяти сохранен в каталоге настройки пользователя.

## Синхронизация событий катастрофического отказа и миниядер между картами управления

`crashlog` уникален для каждой из карт управления, поэтому если катастрофический отказ произойдет, когда карта "8" будет активна, то это будет зарегистрировано на карте "8". Последующее переключение больше не отображало бы катастрофический отказ в журнале. Для получения этого катастрофического отказа коммутатор назад для прочесывания "8" должен быть сделан. Журнал событий катастрофического отказа и дампы уникальны для активных и резервных карт управления, поэтому если катастрофический отказ произойдет на активной карте тогда, то журнал событий катастрофического отказа и отнесенные дампы будут сохранены на активной карте только. Эти сведения об аварийном отказе не доступны на резервной карте. Каждый раз, когда карты переключаются из-за катастрофического отказа в активной карте, и сведения об аварийном отказе больше не отображаются на карте, которая вступает во владение, сведения об аварийном отказе могут быть получены только из текущей активной карты. Для получения списка катастрофического отказа другой карты переключатель требуется снова. Во избежание этого переключателя и получить сведения об аварийном отказе из резервной карты, синхронизации между двумя картами управления и `maintainence` последних сведений об аварийном отказе требуется.

Поступающее событие катастрофического отказа будет передано резервному SMC/MIO и сохранено в `crashlog` файле резерва подобным способом. Миниядро, NPU или дампы ядра на флэш-памяти активного SMC/MIO должен синхронизироваться с резервным SMC/MMIO с `rsync` командой. Когда `crashlog` запись или целый список удалены через команду CLI, это должно быть стерто и на активном и на резервном SMCs/MIOs. Нет никакого влияния на

память. Весь катастрофический отказ отнесся, действие синхронизации будет сделано evlogd резервной карты SMC/MIO, поскольку резерв evlogd менее загружен, и резервная карта имеет достаточно комнаты для действия синхронизации. Поэтому на производительность системы не будут влиять.

## Команды

Эти команды могут использоваться для решения проблем:

```
#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
```

Corefiles генерируются после катастрофического отказа. Обычно операторы хранят их во внешнем сервере. Название corefile обычно похоже на катастрофический отказ - <Cardnum> - <Цифра ЦП> - <Шестнадцатеричная метка времени>-coree.gcrash-09-00-5593a1b8-core.

Каждый раз, когда катастрофический отказ происходит, эти сведения об аварийном отказе сохранены:

- Запись события сохранена в/flash/crashlog2 файле (крешлог).
- Связанное миниядро, NPU или файл дампа ядра сохранены в/flash/crsh2 каталоге.

## Сводка

Все программное обеспечение ASR5x00 разработано для обработки и предсказанных условий/событий и непредвиденных условий/событий. В то время как Cisco стремится иметь совершенное программное обеспечение, неизбежно ошибки будут существовать, и сбои будут возможны. Именно поэтому функция восстановления сеанса так важна. Cisco борется за совершенство, минимизирует вхождения сбоев и откроет сеанс восстановления, позволит сеансам продолжаться после катастрофического отказа. Тем не менее, важно, чтобы Cisco продолжила стремиться достигнуть совершенного программного обеспечения. Меньше сбоев уменьшит вероятность множественных сбоев, которые происходят одновременно. В то время как восстановление сеанса эффективно излечивает одиночный катастрофический отказ, восстановление после множественных одновременных сбоев разработано немного по-другому. Операторы редко должны (или никогда) испытывают множественные одновременные сбои, но если такой должны были произойти, ASR5x00 разработан для восстановления целостности системы как наивысшего приоритета, возможно в жертве некоторых сеансов абонента.