

Fehlerbehebung bei Punt-Keepalive-Fehlern in Cisco IOS XE

Inhalt

[Einleitung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[Hintergrundinformationen](#)

[Die Punt-Debug-Protokolldatei](#)

[Die Linux Shared Memory Punt-Schnittstelle \(LSMPI\)](#)

[Die Punt-Überwachung](#)

[Embedded Event Manager \(EEM\) für die Datensammlung](#)

[Ein praktisches Beispiel](#)

[Verbesserungen](#)

Einleitung

In diesem Dokument wird beschrieben, wie Sie Fehler beim Aufrechterhalten des aktiven Punts beheben.

Voraussetzungen

Anforderungen

Grundkenntnisse von Cisco IOS® XE

Verwendete Komponenten

Dieses Dokument basiert auf Cisco IOS XE-Routern wie den Serien CSR8000v, ASR1000 und ISR4000.

Die Informationen in diesem Dokument beziehen sich auf Geräte in einer speziell eingerichteten Testumgebung. Alle Geräte, die in diesem Dokument benutzt wurden, begannen mit einer gelöschten (Nichterfüllungs) Konfiguration. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die möglichen Auswirkungen aller Befehle kennen.

Hintergrundinformationen

Bei Cisco IOS XE-basierten Systemen ist der Pfad für jeden Eintrag ein interner Datenpfad. Hierbei handelt es sich um den Pfad, über den die Kommunikation zwischen der Kontroll- und

Datenebene stattfindet.

Dieser interne Pfad wird zur Übertragung von Paketen auf der Kontrollebene für die Router-Nutzung verwendet.

Wenn dieser Pfad fehlschlägt, können Sie diese Art von Fehler im Protokoll sehen.

```
%IOSXE_INFRA-4-NO_PUNT_KEEPLIVE: Keepalive not received for 60 seconds
```

Bei den Keep-Alive-Nachrichten handelt es sich um Nachrichten, die den Zustand des Pfads zwischen dem QFP und dem RP überwachen.

Dieser Pfad ist für den Betrieb des Systems von entscheidender Bedeutung.

Wenn diese Keepalives nicht innerhalb von 5 Minuten empfangen werden, können Sie ein kritisches Protokoll wie dieses sehen:

```
%IOSXE_INFRA-2-FATAL_NO_PUNT_KEEPLIVE: Keepalive not received for 300 seconds resetting
```

Das System wird zurückgesetzt, um sich von diesem Zustand zu erholen.

Die Punt-Debug-Protokolldatei

Im Falle von Punt-Keep-Alive-Fehlern und daraus resultierenden Rücksetzvorgängen erstellt das System eine Datei mit dem Namen `punt_debug.log`, die relevante Daten sammelt, um das Verhalten zur Problemzeit zu verstehen.



Anmerkung: Stellen Sie sicher, dass das System mit der neuesten Cisco IOS XE-Softwareversion auf dem neuesten Stand ist, damit die Datei `punt_debug.log` generiert werden kann.

Diese Datei enthält diese Befehle, die mehrmals ausgeführt werden, um verschiedene Zähler zu verstehen.

```
show platform software infra punt-keepalive
```

```
show platform software infra lsmpi
```

```
show platform software infrastructure lsmpi driver
```

```
show platform software infra lsmpi bufusage
```

```
show platform software punt-policer
```

```
show platform software status control-prozessor brief
```

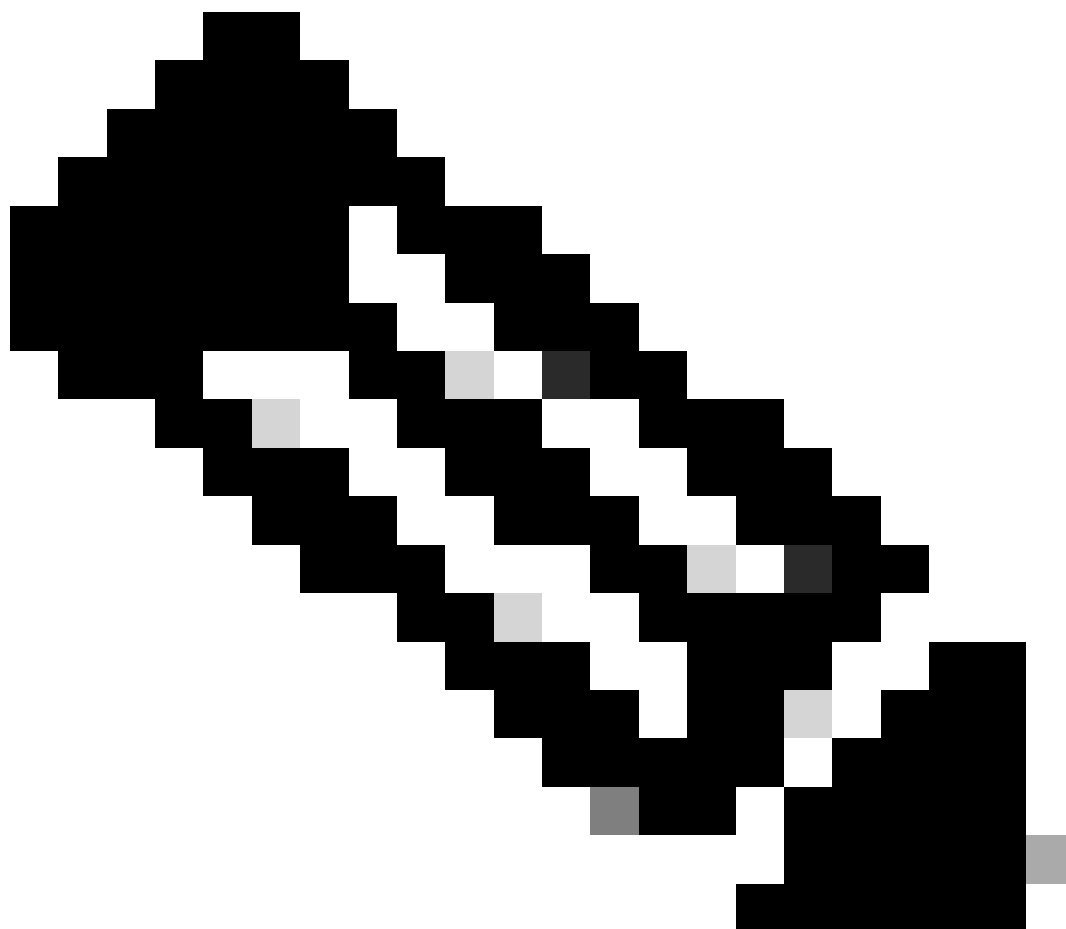
show prozess cpu plattform sortiert

show platform software infrastructure punt

show plattform hardware qfp aktive statistik drop

show plattform hardware qfp active infra punt statistik typ pro ursache

show plattform hardware qfp aktive infrastruktur bqs warteschlange ausgabe standard all



Anmerkung: In der Datei punt_debug.log konzentrieren Sie sich auf Fehleranzeigen und große Paketmengen, die das Problem verursachen können.

Die Linux Shared Memory Punt-Schnittstelle (LSMPI)

Diese Komponente dient zum Übertragen von Paketen und Nachrichten vom Weiterleitungsprozessor an den Routing-Prozessor.

Die Punt-Überwachung

Die Point Policer-Funktion ist ein Schutzmechanismus auf Kontrollebene, mit dem das System Pakete auf Kontrollebene schützen und überwachen kann.

Mit dem Befehl `show platform software punt-policer` können Sie die konformen Pakete und die aufgrund dieses Policers verworfenen Pakete anzeigen.

```
----- show platform software punt-policer -----
```

Per Punt-Cause Policer Configuration and Packet Counters

Punt Cause	Description	Config Rate(pps)		Conform Packets		Dropped Pack
		Normal	High	Normal	High	Normal
2	IPv4 Options	874	655	0	0	0
3	Layer2 control and legacy	8738	2185	0	0	0
4	PPP Control	437	1000	0	0	0

-- snip : output omitted for brevity --

Der Befehl `show platform software infrastructure punt` zeigt Zählerdaten zu punt-Ursachen an.

```
----- show platform software infrastructure punt -----
```

LSMPI interface internal stats:

enabled=0, disabled=0, throttled=0, unthrottled=0, state is ready

Input Buffers = 51181083

Output Buffers = 51150283

-- snip : output omitted for brevity --

EPC CP RX Pkt cleansed 0

Punt cause out of range 0

IOSXE-RP Punt packet causes:

3504959 ARP request or response packets

27 Incomplete adjacency packets

-- snip : output omitted for brevity --

FOR_US Control IPv4 protocol stats:

2369262 TCP packets

FOR_US Control IPv6 protocol stats:

6057 ICMPV6 packets

Packet histogram(500 bytes/bin), avg size in 119, out 95:

Pak-Size	In-Count	Out-Count
0+:	51108211	51144723
500+:	22069	2632
1000+:	2172	0
1500+:	3170	0

Diese Daten sind relevant, um zu verstehen, was sich auf den Punt-Keep-Alive-Pfad auswirken kann.

Embedded Event Manager (EEM) für die Datensammlung

Falls die Datei `punt_debug.log` nicht genügend Daten zur Diagnose des Problems liefert, kann EEM-Scripting verwendet werden, um mehr Datenpunkte zur Problemzeit zu erhalten.

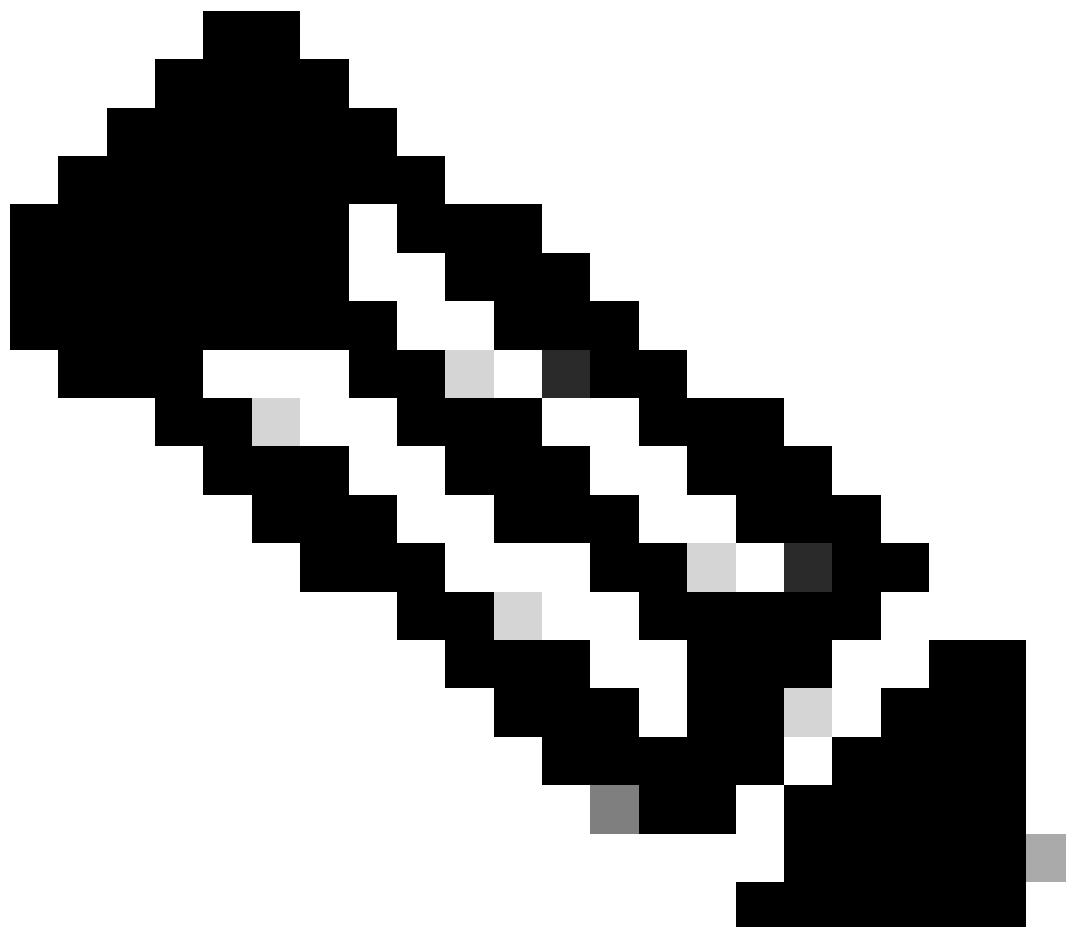
```
event manager applet punt_script authorization bypass
event syslog pattern "IOSXE_INFRA-4-NO_PUNT_KEEPLIVE" maxrun 1000
action 0.0 cli command "enable"
action 0.1 set i "0"
action 0.2 cli command "test platform software punt-keepalive ignore-fault"
action 0.3 while $i lt 10
action 0.4 syslog msg "iteration $i"
action 0.9 cli command "show clock | append bootflash:qfp_lsmpi.txt"
action 1.0 cli command "show platform software infrastructure lsmpi | append bootflash:qfp_lsmpi.txt"
action 1.1 cli command "show platform software infrastructure lsmpi driver | append bootflash:qfp_lsmpi.txt"
action 1.2 cli command "show platform software infrastructure lsmpi driver 0 | append bootflash:qfp_lsmpi.txt"
action 1.3 cli command "show platform software infrastructure lsmpi bufusage | append bootflash:qfp_lsmpi.txt"
action 1.4 cli command "show platform software infrastructure lsmpi bufusage 0 | append bootflash:qfp_lsmpi.txt"
action 1.5 cli command "show platform software infrastructure punt-keepalive | append bootflash:qfp_lsmpi.txt"
action 1.6 cli command "show platform software infrastructure punt | append bootflash:qfp_lsmpi.txt"
action 1.7 cli command "show platform software punt-policer | append bootflash:qfp_lsmpi.txt"
action 1.8 cli command "show platform hardware qfp active infrastructure punt stat type per-cause | append bootflash:qfp_lsmpi.txt"
action 1.9 cli command "show platform hardware qfp active infrastructure punt statistics type punt-drop | append bootflash:qfp_lsmpi.txt"
action 1.a cli command "show platform hardware qfp active infrastructure punt statistics type inject-drop | append bootflash:qfp_lsmpi.txt"
action 1.b cli command "show platform hardware qfp active infrastructure bqs queue output default interface | append bootflash:qfp_lsmpi.txt"
action 1.c cli command "show platform hardware qfp active statistics drop | append bootflash:qfp_lsmpi.txt"
action 1.d cli command "show platform hardware qfp active datapath utilization | append bootflash:qfp_lsmpi.txt"
action 1.e cli command "show platform hardware qfp active datapath infrastructure sw-hqf | append bootflash:qfp_lsmpi.txt"
action 1.f cli command "show platform hardware qfp active datapath infrastructure sw-distrib | append bootflash:qfp_lsmpi.txt"
action 1.g cli command "show platform hardware qfp active datapath infrastructure sw-pktmem | append bootflash:qfp_lsmpi.txt"
action 1.h cli command "show platform software status control-processor brief | append bootflash:qfp_lsmpi.txt"
action 2.0 increment i
action 2.1 wait 3
action 2.4 end
action 3.0 syslog msg "End of data collection. Please transfer the file at bootflash:qfp_lsmpi.txt"
action 5.0 cli command "debug platform hardware qfp active datapath crashdump"
```



Anmerkung: Die im Skript enthaltenen Befehle variieren je nach Plattform, auf der es konfiguriert ist.

Mit diesem Skript können Sie den LSMPI-, Ressourcen- und Punt-Status während der Problemzeit verstehen.

Das EEM-Skript enthält den Befehl `debug platform hardware qfp active datapath crashdump`, der das vom Entwicklerteam und TAC benötigte QFP-Core-Dump generiert.



Anmerkung: Wenn Sie beim Cisco TAC einen Fall einreichen, geben Sie die vom Skript generierte Kerndatei an.

Wenn eine Paketverfolgung erforderlich ist, kann diese Änderung dem Skript hinzugefügt werden:

Richten Sie zunächst die Paketablaufverfolgungskonfiguration ein, die über das EEM-Skript ausgeführt werden kann:

```
debug platform packet-trace paket 8192 fia-trace zirkular  
Debug-Plattformbedingung  
debug platform packet-trace copy paket beide L2
```

Starten und stoppen Sie es anschließend mit den folgenden Aktionen im EEM-Skript:

```
action 6.2 cli-Befehl "debug platform condition start"  
Aktion 6.3 warten 8  
action 6.4 cli-Befehl "debug platform condition stop"
```


Speichern Sie die Daten mit diesen Befehlen in einer separaten Datei:

```
action 6.5 cli-Befehl "show platform packet trace statistics" | append bootflash:traceAll.txt"
```

```
action 6.6 cli-Befehl "show platform packet-trace summary | append bootflash:traceAll.txt"
```

```
action 6.7 cli-Befehl "show platform packet-trace packet all decode | append bootflash:traceAll.txt"
```

Diese Paketablaufverfolgungsaktionen werden direkt nach der end-Anweisung des while-Zyklus innerhalb des EEM-Skripts hinzugefügt.

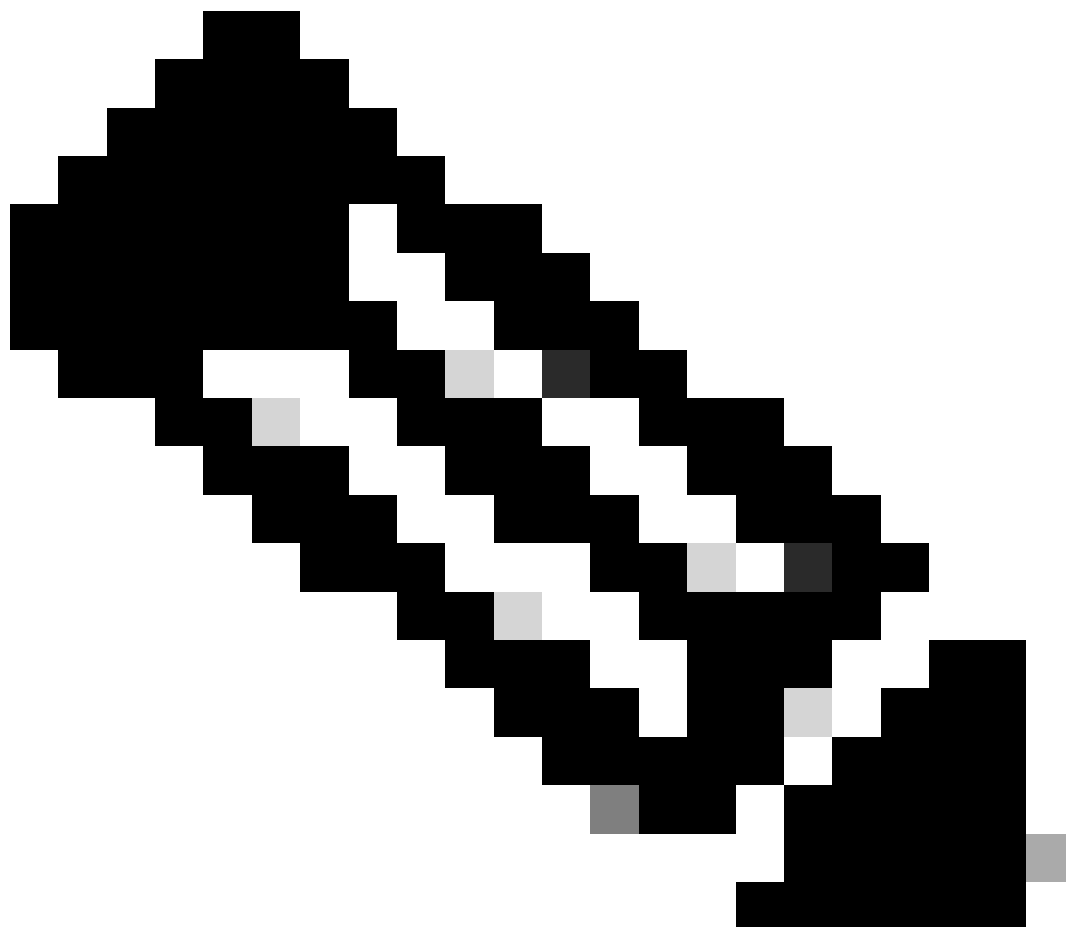
Mit diesem Skript können Sie ermitteln, welche Art von Paketen das Problem verursachen können.

Die Paketverfolgung ist eine Funktion, die bei der [Fehlerbehebung mit der IOS XE DataPath Packet Trace-Funktion](#) dokumentiert ist.

Ein praktisches Beispiel

Ein CSR8000v wird ständig neu gestartet.

Nachdem Sie den Systembericht extrahiert haben, können Sie einen Crashdump und eine IoSD-Kerndatei beobachten, die darauf hinweist, dass die entsprechenden Funktionen im Stapelverlauf beibehalten werden.



Anmerkung: Für die Stack-Trace-Decodierung ist die Unterstützung des TAC erforderlich.

Die crashinfo-Datei ist jedoch in Klartext, und Sie können die folgenden Symptome sehen:

```
Jan 15 14:29:41.756 AWST: %IOSXE_INFRA-4-NO_PUNT_KEEPALIVE: Keepalive not received for 160 seconds
Jan 15 14:30:01.761 AWST: %IOSXE_INFRA-4-NO_PUNT_KEEPALIVE: Keepalive not received for 180 seconds
Jan 15 14:30:21.766 AWST: %IOSXE_INFRA-4-NO_PUNT_KEEPALIVE: Keepalive not received for 200 seconds
Jan 15 14:30:41.776 AWST: %IOSXE_INFRA-4-NO_PUNT_KEEPALIVE: Keepalive not received for 220 seconds
Jan 15 14:31:01.780 AWST: %IOSXE_INFRA-4-NO_PUNT_KEEPALIVE: Keepalive not received for 240 seconds
Jan 15 14:31:41.789 AWST: %IOSXE_INFRA-4-NO_PUNT_KEEPALIVE: Keepalive not received for 280 seconds
Jan 15 14:32:01.791 AWST: %IOSXE_INFRA-4-NO_PUNT_KEEPALIVE: Keepalive not received for 300 seconds
Jan 15 14:32:01.791 AWST: %IOSXE_INFRA-2-FATAL_NO_PUNT_KEEPALIVE: Keepalive not received for 300 seconds
```

%Software-forced reload

Exception to IOS Thread:

Frame pointer 0x7F0AE0EE29A8, PC = 0x7F0B342C16D2

UNIX-EXT-SIGNAL: Aborted(6), Process = PuntInject Keepalive Process

Verbesserungen

Ab der Version Cisco IOS XE 17.15 über die Cisco Bug-ID [CSCwf85505](#) wurde eine Erweiterung für die automatische Generierung von QFP-Core-Dateien eingeführt.

Informationen zu dieser Übersetzung

Cisco hat dieses Dokument maschinell übersetzen und von einem menschlichen Übersetzer editieren und korrigieren lassen, um unseren Benutzern auf der ganzen Welt Support-Inhalte in ihrer eigenen Sprache zu bieten. Bitte beachten Sie, dass selbst die beste maschinelle Übersetzung nicht so genau ist wie eine von einem professionellen Übersetzer angefertigte. Cisco Systems, Inc. übernimmt keine Haftung für die Richtigkeit dieser Übersetzungen und empfiehlt, immer das englische Originaldokument (siehe bereitgestellter Link) heranzuziehen.