Risoluzione dei problemi relativi agli errori Punt Keepalive in Cisco IOS XE

Sommario

Introduzione

Questo documento descrive come risolvere i problemi relativi agli errori di punt keep alive.

Prerequisiti

Requisiti

Conoscenze base di Cisco IOS® XE.

Componenti usati

Questo documento si basa sui router Cisco IOS XE come la serie CSR800v, ASR1000 e ISR4000.

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

Premesse

Nei sistemi Cisco IOS XE, il percorso punt è un percorso di dati interno. Si tratta del percorso in cui avviene la comunicazione tra il piano di controllo e il piano dati.

Questo percorso interno viene utilizzato per trasmettere i pacchetti del control plane per il consumo del router.

Se il percorso non riesce, questo tipo di errore viene visualizzato nel registro.

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

I messaggi keep-alive sono messaggi che monitorano lo stato del percorso tra QFP e RP.

Questo percorso è fondamentale per il funzionamento del sistema.

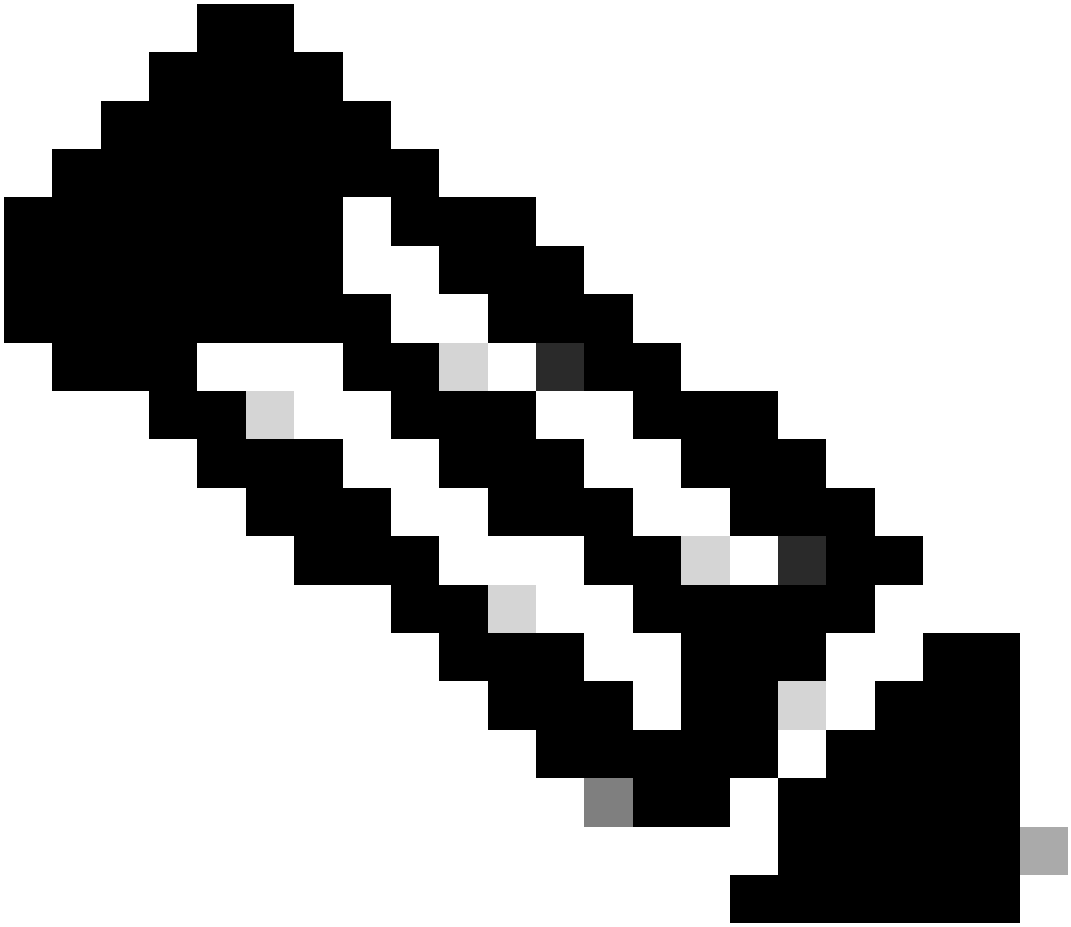
Se i messaggi di conservazione non vengono ricevuti entro 5 minuti, è possibile visualizzare un registro critico simile al seguente:

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

Il sistema viene reimpostato per ripristinare la configurazione originale.

File di registro di debug Punt

In caso di guasti e reimpostazioni dovuti al punt keep-alive, il sistema crea un file denominato punt_debug.log che raccoglie i dati rilevanti per comprendere il comportamento in fase di emissione.



Nota: Accertarsi di avere il sistema aggiornato con l'ultima versione del software Cisco IOS XE per generare il file punt_debug.log.

Questo file contiene l'esecuzione di questi comandi più volte per comprendere i diversi contatori.

show platform software infra punt-keepalive

show platform software infra lsmpi

show platform software infrastructure driver lsmpi

show platform software infra lsmpi bufusage

show platform software punt-policer

show platform software status control-processor brief

mostra piattaforma cpu processo ordinata

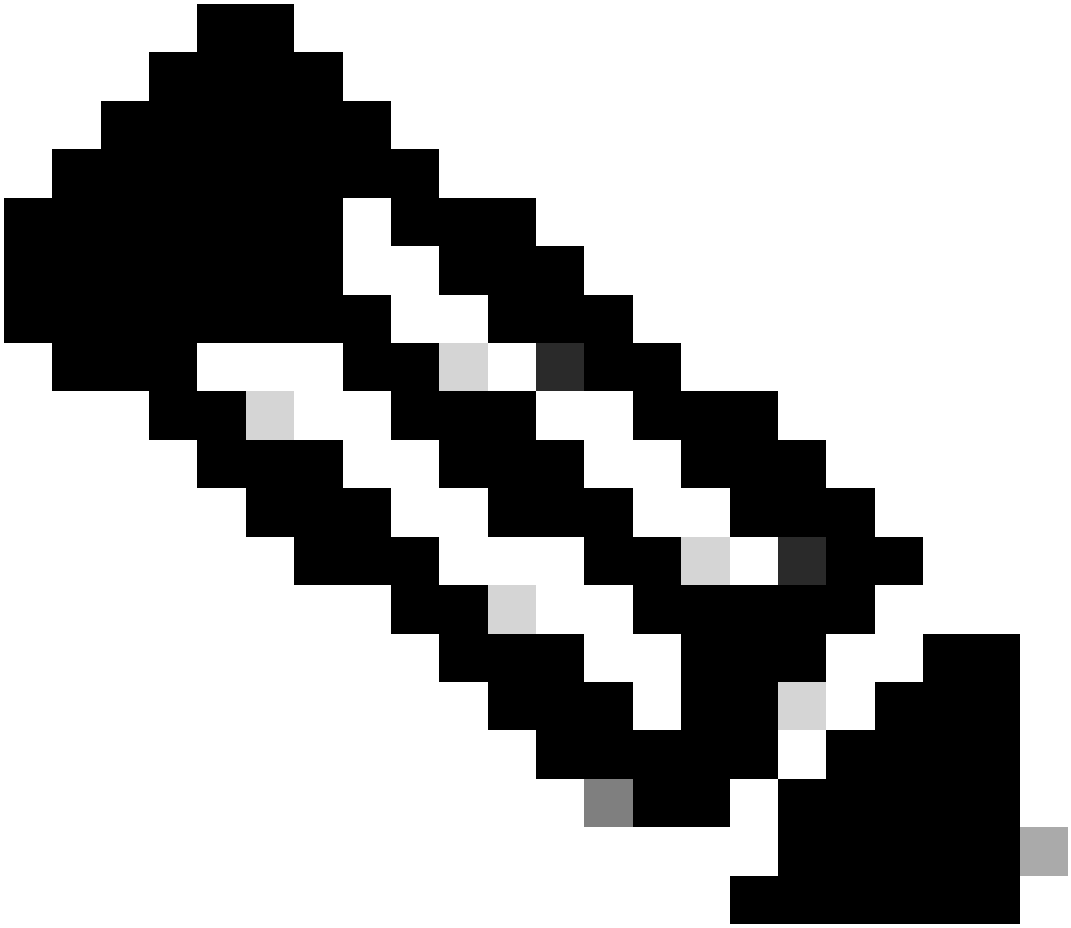
show platform software infrastructure punt

show platform hardware qfp active statistics drop

show platform hardware qfp active statistiche punt a infrarossi per causa

show platform hardware qfp active infrastructure bqs coda output predefinito tutto

---



Nota: All'interno di punt_debug.log, ci si concentra sugli indicatori di errore e sulla grande quantità di pacchetti che possono causare il problema.

Interfaccia LSMPI (Linux Shared Memory Punt Interface)

Questo componente viene utilizzato per trasmettere pacchetti e messaggi dal processore di inoltro al processore di routing.

Punt Policer

Il punt policer è un meccanismo di protezione control plane che consente al sistema di proteggere e controllare i pacchetti control plane.

Con il comando show platform software punt-policer, è possibile visualizzare i pacchetti conform e i pacchetti scartati a causa di questo policer.

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


Per Punt-Cause Policer Configuration and Packet Counters


Punt                                Config Rate(pps)     Conform Packets                  Dropped Packe
Cause  Description                   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  --
```

Il comando show platform software infrastructure punt mostra i dati del contatore sulle cause dei punt.

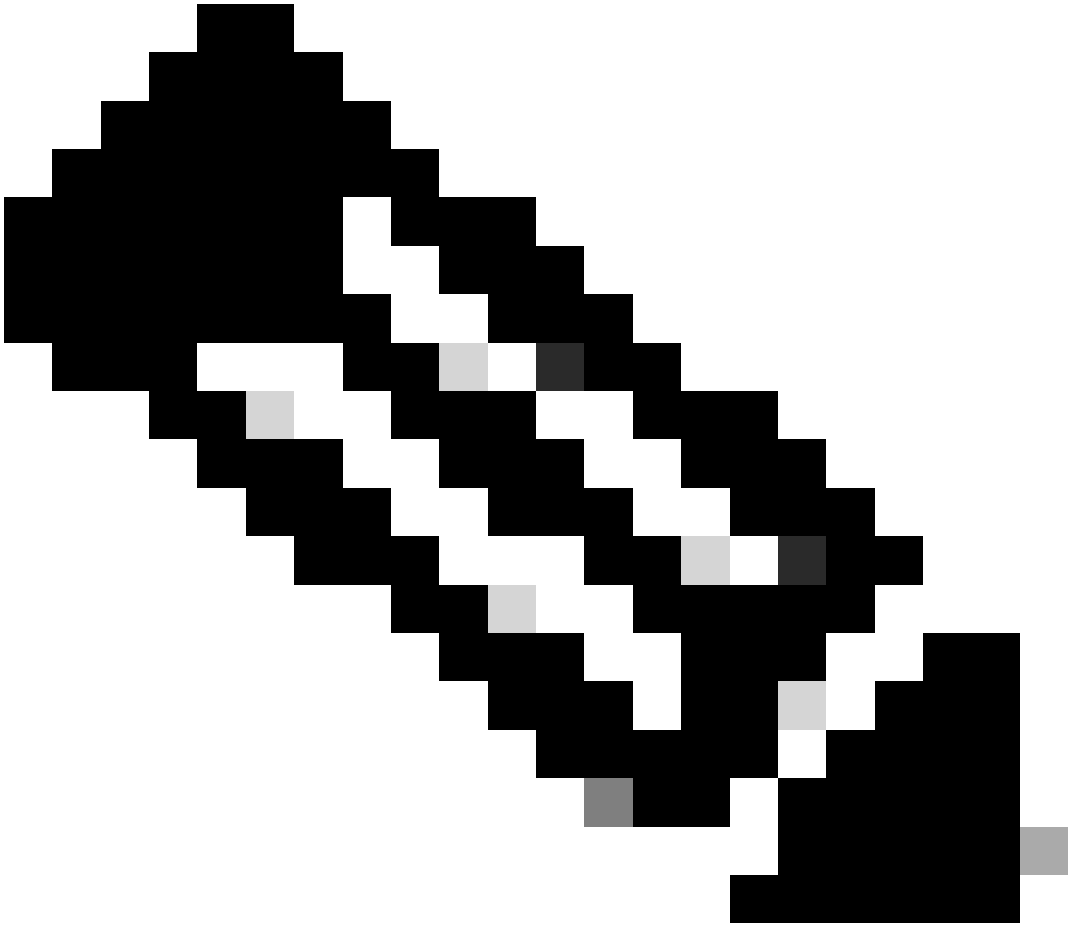
```
------------------ show platform software infrastructure punt ------------------


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 protcol stats:
        2369262 TCP packets

   FOR_US Control IPv6 protcol 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
```

Questi dati sono importanti per capire quali conseguenze possono avere sul percorso punt keep alive.

Embedded Event Manager (EEM) per la raccolta dati

Nel caso in cui il file punt_debug.log non fornisca dati sufficienti per diagnosticare il problema, è possibile utilizzare gli script EEM per ottenere più punti dati in fase di emissione.

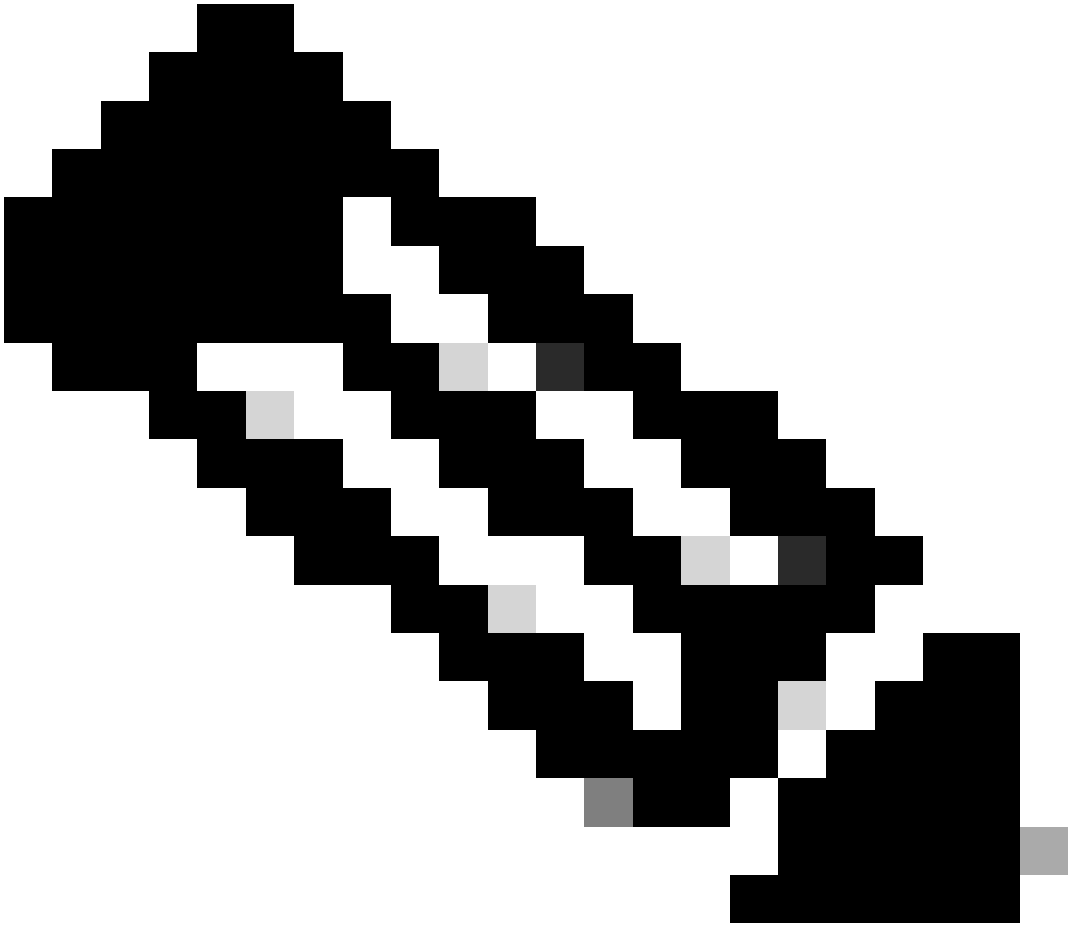
```
event manager applet punt_script authorization bypass
event syslog pattern "IOSXE_INFRA-4-NO_PUNT_KEEPALIVE" maxrun 1000
action 0.0 cli command "enable"
action 0.1 set i "0"
action 0.2 cli command "test platform software punt-keepalive ignore-fault"
action 0.3 while $i lt 10
action 0.4 syslog msg "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
action 1.2 cli command "show platform software infrastructure lsmpi driver 0 | append bootflash:qfp_lsm
action 1.3 cli command "show platform software infrastructure lsmpi bufusage | append bootflash:qfp_lsm
action 1.4 cli command "show platform software infrastructure lsmpi bufusage 0 | append bootflash:qfp_l
action 1.5 cli command "show platform software infrastructure punt-keepalive | append bootflash:qfp_lsm
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 | app
action 1.9 cli command "show platform hardware qfp active infrastructure punt statistics type punt-drop
action 1.a cli command "show platform hardware qfp active infrastructure punt statistics type inject-dr
action 1.b cli command "show platform hardware qfp active infrastructure bqs queue output default inter
action 1.c cli command "show platform hardware qfp active statistics drop | append bootflash:qfp_lsmpi.
action 1.d cli command "show platform hardware qfp active datapath utilization | append bootflash:qfp_l
action 1.e cli command "show platform hardware qfp active datapath infrastructure sw-hqf | append bootf
action 1.f cli command "show platform hardware qfp active datapath infrastructure sw-distrib | append b
action 1.g cli command "show platform hardware qfp active datapath infrastructure sw-pktmem | append bo
action 1.h cli command "show platform software status control-processor brief | append bootflash:qfp_ls
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"
```



Nota: I comandi contenuti nello script variano a seconda della piattaforma in cui è configurato.

Questo script consente di comprendere lo stato lsmpi, risorse e punt durante il tempo di emissione.

Lo script EEM include il comando debug platform hardware qfp active datapath crashdump che genera il dump del core qfp, richiesto dal team di sviluppatori e da TAC.



Nota: Se si presenta una richiesta a Cisco TAC, fornire il file principale generato dallo script.

Se è necessaria una traccia del pacchetto, è possibile aggiungere allo script la seguente modifica:

Innanzitutto, configurare la configurazione della traccia del pacchetto, che può essere eseguita dallo script EEM:

debug platform packet-trace packet 8192 fi-trace circolare
debug platform condition both
debug platform packet-trace copy packet entrambi L2

Quindi, avviarlo e interromperlo con queste azioni all'interno dello script EEM:

azione 6.2 comando cli "debug platform condition start"
azione 6.3 attendere 8
azione 6.4 comando cli "debug platform condition stop"

Eseguire quindi il dump dei dati con questi comandi in un file separato:

azione 6.5 comando cli "show platform packet-trace statistics | append bootflash:traceAll.txt"
azione 6.6 comando cli "show platform packet-trace summary | append bootflash:traceAll.txt"
azione 6.7 comando cli "show platform packet-trace packet all decode | append bootflash:traceAll.txt"

La logica delle azioni di traccia del pacchetto viene aggiunta subito dopo l'istruzione end del ciclo while all'interno dello script EEM.

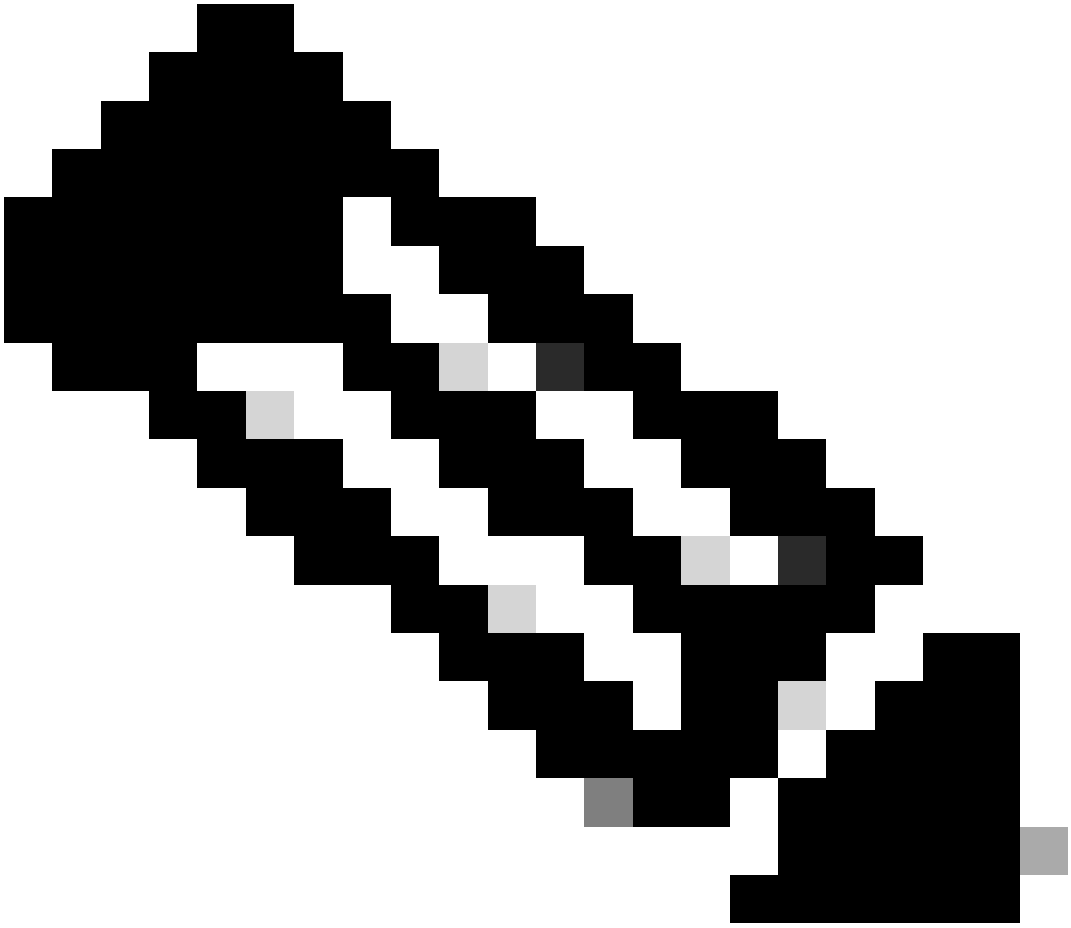
Questo script consente di comprendere il tipo di pacchetti che possono causare il problema.

Packet trace è una funzione documentata in Risoluzione dei problemi relativi a IOS XE Datapath Packet Trace Feature

Un esempio pratico

Un CSR800v viene riavviato costantemente.

Dopo aver estratto il report di sistema, è possibile osservare un'immagine di crash dump e un file di base iosd che indica che le funzioni correlate a punt keep-alive sono mantenute attive all'interno dello stack trace.



Nota: Per la decodifica della traccia dello stack, è richiesta l'assistenza TAC.

Tuttavia, il file crashinfo è in formato testo non crittografato ed è possibile visualizzare i seguenti sintomi:

```
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 secon

%Software-forced reload

Exception to IOS Thread:
Frame pointer 0x7F0AE0EE29A8, PC = 0x7F0B342C16D2
```

```
UNIX-EXT-SIGNAL: Aborted(6), Process = PuntInject Keepalive Process
-Traceback= 1#7b5996c3
```

Il processo interessato è il processo PuntInject Keepalive.

Il sistema deve attivare un segnale di interruzione quando il comando keepalive raggiunge il valore di soglia di 300 secondi.

Il file punt_debug.log rivela alcuni errori di trasmissione nel comando show platform software infrastructure lsmpi driver:

```
Reason for TX drops (sticky):
    Bad packet len   : 0
    Bad buf len      : 0
    Bad ifindex      : 0
    No device        : 0
    No skbuff        : 0
    Device xmit fail : 82541   >>>>>>>>>>>>>>>>>>>> Tx failure
```

Si tratta di un fallimento generico.

Questo contatore aumenta se nel file sono stati prelevati più campioni.

Lo script EEM è stato fornito per ottenere ulteriori dati sulle risorse, sul percorso dei dati punt e su altri comandi relativi all'infrastruttura.

Controllando i contatori lsmpi traffic punt si nota che i pacchetti del control plane EIGRP sono notevoli. Si tratta di pacchetti identificati come per i pacchetti USA:

```
17660574 For-us data packets
543616 RP<->QFP keepalive packets
1004 Glean adjacency packets
3260636 BFD control packets
122523839 For-us control packets<<<<

FOR_US Control IPv4 protcol stats:
153551 TCP packets
2663105 GRE packets
104394559 EIGRP packets<<<<
```

In seguito, è stato rilevato che l'hypervisor aveva una sottoscrizione eccessiva, con un impatto sulle risorse di elaborazione sottostanti.

Il CSR8000v è stato implementato in un altro hypervisor, contribuendo a ridurre il problema.

Miglioramenti

Sono stati introdotti miglioramenti per la generazione automatica di file core qfp a partire dalla versione Cisco IOS XE 17.15 con ID bug Cisco CSCwf85505

Informazioni su questa traduzione

Cisco ha tradotto questo documento utilizzando una combinazione di tecnologie automatiche e umane per offrire ai nostri utenti in tutto il mondo contenuti di supporto nella propria lingua. Si noti che anche la migliore traduzione automatica non sarà mai accurata come quella fornita da un traduttore professionista. Cisco Systems, Inc. non si assume alcuna responsabilità per l’accuratezza di queste traduzioni e consiglia di consultare sempre il documento originale in inglese (disponibile al link fornito).