Belangrijke informatie over debug-opdrachten

Inhoud

Inleiding

Deze pagina biedt een aantal algemene richtlijnen voor het gebruik van de beschikbare apparaten op Cisco IOS® platforms, evenals voorbeelden voor het correct gebruik van de **debug ip packet** commando en voorwaardelijke debugging.

> **Opmerking:** Dit document legt niet uit hoe u specifieke debug-opdrachten en -uitgangen kunt gebruiken en interpreteren. Raadpleeg de juiste documentatie voor de opdracht Debug van Cisco voor informatie over specifieke informatie **debug** opdrachten.

De output van **debug** geprivilegieerde EXEC opdrachten bieden diagnostische informatie die een verscheidenheid aan internetactiviteiten omvat die verband houden met de protocolstatus en netwerkactiviteit in het algemeen.

Voorwaarden

Vereisten

Cisco raadt kennis van de volgende onderwerpen aan:

- Aansluiten op de router met de console-, aux- en vty poorten
- Algemene Cisco IOS-configuratieproblemen
- Cisco IOS-debug-ingangen

Gebruikte componenten

Dit document is niet beperkt tot specifieke software- en hardware-versies.

De informatie in dit document is gebaseerd op de apparaten in een specifieke laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk live is, moet u de potentiële impact van elke opdracht begrijpen.

Waarschuwingen

Gebruik **debug** opdrachten met voorzichtigheid. In het algemeen wordt aanbevolen deze opdrachten alleen te gebruiken onder de richting van uw vertegenwoordiger voor technische ondersteuning van de router wanneer er problemen worden opgelost.

Het in werking stellen van het debuggen kan de werking van de router verstoren wanneer internetwerken hoge belastingsomstandigheden ervaren. Vandaar, als het registreren wordt geactiveerd, kan de toegangsserver met tussenpozen bevriezen zodra de console poort overbelast wordt met logberichten.

Voordat u met een **debug** Bespreek altijd de uitvoer die deze opdracht zal genereren en de hoeveelheid tijd die deze kan duren. Als u bijvoorbeeld een router met één basisinterface (BRI) hebt, **debug isdn q931** waarschijnlijk zal het systeem geen schade berokkenen . Maar het doen van hetzelfde debug op een AS5800 met volledige E1 configuratie kan waarschijnlijk zoveel input genereren dat het kan hangen en stoppen met reageren.

Alvorens te zuiveren, bekijk uw CPU-lading met de **show processes cpu** uit. Controleer dat u meer dan 100 CPU beschikbaar hebt voordat u met de opslagapparaten begint. Raadpleeg het gebruik van hoge CPU's voor probleemoplossing op Cisco-routers voor meer informatie over de manier waarop u ladingen met hoge CPU's kunt verwerken. Als u bijvoorbeeld een Cisco 7200-router hebt met een ATM-interface die overbrugging doet, kan het opnieuw starten van de router, afhankelijk van de hoeveelheid subinterfaces die is geconfigureerd, veel van zijn CPU gebruiken. De reden is dat, voor elk virtueel circuit (VC), een Bridge Protocol Data Unit (BPDU)-pakket moet worden gegenereerd. Het starten van defecten tijdens zo'n kritieke tijd kan ertoe leiden dat het CPU-gebruik dramatisch stijgt en kan resulteren in een verlies van hang of netwerkconnectiviteit.

> **Opmerking:** Wanneer insecten lopen, ziet u gewoonlijk de routerherinnering niet, vooral wanneer het debug intensief is. Maar in de meeste gevallen, kunt u de no gebruiken om alle of undebug alle opdrachten te gebruiken om de uitwerpselen te stoppen. Raadpleeg het gedeelte Debug Outputs verkrijgen voor meer informatie over veilig gebruik van apparaten.

Conventies

Raadpleeg Cisco Technical Tips Conventions (Conventies voor technische tips van Cisco) voor meer informatie over documentconventies.

Voordat u het programma gaat reinigen

Controleer, naast de hierboven genoemde punten, of u de impact van de uitbarstingen op de stabiliteit van het platform begrijpt. U dient ook te overwegen op welke interface op de router u een verbinding moet maken. Dit deel bevat enkele richtsnoeren.

Debug Outputs verkrijgen

Routers kunnen debug uitvoer naar verschillende interfaces weergeven, waaronder de console-, aux- en vty poorten. De routers kunnen ook berichten naar een interne buffer op een externe unix syslogserver loggen. Instructies en voorbehouden voor elke methode worden hieronder besproken:

console-poort

Als u in normale configuraties op de console bent aangesloten, hoeft er geen extra werk te worden gedaan. De debug-uitvoer moet automatisch worden weergegeven. Maar zorg dat het **logging console level** is ingesteld op de gewenste waarde en dat houtkap niet is uitgeschakeld met de **no logging console** uit.

> Waarschuwing: De buitensporige tekorten aan de troostpoort van een router kunnen het veroorzaken om te hangen. Dit is omdat de router automatisch de console uitvoer voorafgaand aan andere routerfuncties prioriteert. Vandaar dat als de router een grote debug uitvoer naar de troostpoort verwerkt, het kan hangen. Als de debug uitvoer buitensporig is, kunt u de vty (telnet) poorten of de logbuffers gebruiken om uw debugs te verkrijgen. Hieronder vindt u meer informatie.

> Opmerking: Standaard is houtkap ingeschakeld op de console-poort. Vandaar dat de console poort de output altijd debug van uitvoer verwerkt zelfs als u eigenlijk een andere poort of methode (zoals Aux, Vty of buffer) gebruikt om de output op te nemen. Vandaar, raadt Cisco aan, onder normale bedrijfsomstandigheden, u het geen houtkapconsole bevel hebt die op elk moment wordt toegelaten en andere methodes gebruikt om Debugs op te nemen. In situaties waar u de console moet gebruiken, draai tijdelijk de houtkap weer in.

Aux-poort

Als u via een extra poort bent verbonden, typt u het **terminal monitor** uit. Controleer ook of **no logging on** Deze opdracht is niet op de router geactiveerd.

> Opmerking: Als u de haven van Aux gebruikt om de router te controleren, houd in gedachten dat, wanneer de router herstart, de haven van Aux niet de uitvoer van de laars van de sequentie toont. Sluit aan op de console poort om de laars volgorde te bekijken.

VTY-poorten

Als u via een extra poort of via telnet bent verbonden, typt u het volgende: **terminal monitor** uit. Controleer ook of **no logging on** commando is niet gebruikt.

Logberichten aan een interne buffer

Het standaard logapparaat is de console; alle berichten worden op de console weergegeven, tenzij anders bepaald.

Om berichten naar een interne buffer te loggen, gebruikt u de **logging buffered** opdracht voor het configureren van de router. Dit is de volledige syntax van deze opdracht:

```
logging buffered
no logging buffered
```

Het **logging buffered** De opdracht kopieert logberichten naar een interne buffer in plaats van ze te schrijven naar de console. De buffer is circulair van aard, dus nieuwere berichten overschrijven oudere berichten. Om de berichten weer te geven die in de buffer zijn inlogd, gebruikt u de geprivilegieerde EXEC-opdracht **show logging**. Het eerste bericht dat wordt weergegeven, is het oudste bericht in de buffer. U kunt de grootte van de buffer en het ernst van de te registreren berichten specificeren.

> **Tip**: Zorg ervoor dat er genoeg geheugen in het vakje aanwezig is voordat u de buffergrootte invoert. Gebruik Cisco IOS **show proc mem** opdracht om geheugen beschikbaar te zien.

Het **no logging buffered** De opdracht annuleert het gebruik van de buffer en schrijft berichten naar de console (de standaard).

Vastlegging van berichten naar een UNIX-bladeserver

Om berichten aan de syslogserver host te loggen, gebruikt u de opdracht voor de configuratie van de logrouter. De volledige syntaxis van deze opdracht is als volgt:

```
logging no logging
```

Het **logging** De opdracht identificeert een syslogserver host om logingberichten te ontvangen. Het argument **<** *ip-adres***>** is het IP-adres van de host. Door deze opdracht meer dan eens uit te geven, maakt u een lijst met syslogservers die houtkapberichten ontvangen.

Het **no logging** De opdracht verwijdert de syslogserver met het opgegeven adres uit de lijst met syslogs.

Andere taken vóór het reinigen

1. Stel de software van de eindemulator in (bijvoorbeeld HyperTerminal) zodat deze de debug uitvoer naar een bestand kan opnemen. In HyperTerminal klikt u bijvoorbeeld op **Transfer**, dan klikt u op **Capture Text**en kies de juiste opties. Raadpleeg voor meer informatie het opnemen van tekstuitvoer uit hyperterminal. Raadpleeg voor andere terminale emulator-software de softwaredocumentatie.
2. timestamps inschakelen van milliseconde (msec) met behulp van de **service timestamps** opdracht:
   ```
   router(config)#service timestamps debug datetime msec
   router(config)#service timestamps log datetime msec
   ```

Deze opdrachten voegen tijdstempels toe aan uiteinden in het formaat MMM DD HH:MM:SS, wat de datum en de tijd aangeeft volgens de systeemklok. Als de systeemklok niet is ingesteld, worden de datum en de tijd voorafgegaan door een sterretje (*) om aan te geven dat de datum en de tijd waarschijnlijk niet correct zijn.

Het is over het algemeen raadzaam om milliseconde tijdstempels te configureren aangezien dit een hoog niveau van helderheid biedt wanneer het kijken naar debug output. De tijden van de Milliseconden zijn een betere indicatie van de timing van de verschillende gebeurtenissen van de debugs in vergelijking met elkaar. Houd er echter rekening mee dat wanneer de console poort veel berichten uitslaat, ze niet correleren met de actuele timing van de gebeurtenis. Bijvoorbeeld, als u toelaat **debug x25** alles op een vakje met 200 VC's en de uitvoer wordt aangemeld bij de buffer (met behulp van de **no logging console** en **logging buffered** opdrachten), is de tijdstempel die in de debug-uitvoer (binnen de buffer) wordt weergegeven, mogelijk niet het juiste tijdstip waarop het pakket door de interface passeert. Gebruik daarom geen msec timestamps om prestatiekwesties aan te tonen, maar om relatieve informatie te verkrijgen over wanneer gebeurtenissen plaatsvinden.

Stoppen met afluisteren

Gebruik de **no debug all** of **undebug all** opdrachten. Controleer of de knoppen zijn uitgeschakeld met de opdracht **show debug**.

Vergeet niet dat de opdrachten **no logging console** en **terminal no monitor** Voorkom alleen dat de uitvoer op de console, Aux of Vty wordt uitgevoerd. Het houdt het debuggen niet op en gebruikt daarom routerbronnen op.

De debug ip-pakketopdracht gebruiken

Het **debug ip packet** opdracht produceert informatie op pakketten die niet snel door de router worden geschakeld. Aangezien het echter een uitvoer voor elk pakje genereert, kan de uitvoer uitgebreid zijn en kan de router daarom ophangen. Om deze reden alleen gebruiken **debug ip packet** onder de in dit punt beschreven strengste controles.

De beste manier om de productie van **debug ip packet** is een toegangslijst te maken die gekoppeld is aan het debug. Alleen pakketten die overeenkomen met de toegangscriteria zijn onderworpen aan **debug ip packet**. Deze toegangslijst hoeft niet op een willekeurige interface te worden toegepast, maar is eerder van toepassing op de debug-handeling.

Voordat u gebruikt **debugging ip packet**Let er op dat de router standaard snel overschakelt of dat hij, indien geconfigureerd, CEF-switching gebruikt. Dit betekent dat, zodra deze technieken zijn geïnstalleerd, het pakje niet aan de processor wordt verstrekt, dus het debuggen geeft niets weer. Om dit te werken, moet u het snel inschakelen van de router uitschakelen met **no ip route-cache** (voor eenastpakketten) of **no ip mroute-cache** (voor multicast pakketten). Dit moet worden toegepast op de interfaces waar het verkeer verondersteld wordt te stromen. Controleer dit bij de **show ip route** uit.

Waarschuwingen:

- Wanneer een router die een groot aantal pakketten verwerkt, snel wordt ingeschakeld, kan het gebruik van CPU's tot pieken leiden, zodat het vak zijn verbinding met de peers hangt of verliest.
- Schakel geen snelswitching uit op een router die Multi Protocol Label Switching (MPLS) uitvoeren. MPLS wordt gebruikt in combinatie met CEF. Daarom kan het in- en uitschakelen van de snelle inschakeling op de interface rampzalige gevolgen hebben.

Laten we een steekproefscenario overwegen:

De access-list ingesteld op router_122 is:

```
access-list 105 permit icmp host 10.10.10.2 host 13.1.1.1
access-list 105 permit icmp host 13.1.1.1 host 10.10.10.2
```

Deze toegangslijst maakt elk protocol van het Internet Control Message Protocol (ICMP)-pakket van host router_121 (met IP-adres 10.10.10.2) naar host router_123 (met IP-adres 13.1.1.1), evenals in de andere richting mogelijk. Het is belangrijk dat u de pakketten in elke richting toestaat, anders kan de router het terugkerende ICMP pakket laten vallen.

Verwijder snel-schakeling op slechts één interface op router_122. Dit betekent dat u slechts de diepten voor de pakketten kunt zien die voor die interface bestemd zijn, zoals gezien vanuit het perspectief van IOS die het pakje onderschept. Van de merken verschijnen zulke pakketten met "d=". Aangezien u nog niet snelle switching op de andere interface hebt uitgeschakeld, is het retourpakket niet onderworpen aan **debug ip packet**. Deze uitvoer toont hoe u snelle omschakeling kunt verhinderen:

```
router_122(config)#interface virtual-template 1
router_122(config-if)#no ip route-cache
router_122(config-if)#end
```

U moet nu activeren **debug ip packet** waarvan de toegangslijst eerder is gedefinieerd (toegangslijst 105).

```
router_122#debug ip packet detail 105
IP packet debugging is on (detailed) for access list 105
router_122#
00:10:01: IP: s=13.1.1.1 (Serial3/0), d=10.10.10.2 (Virtual-Access1),
g=10.10.10.2, len 100, forward

00:10:01:     ICMP type=0, code=0
! -- ICMP packet from 13.1.1.1 to 10.10.10.2. ! -- This packet is displayed because it matches
the ! -- source and destination requirements in access list 105 00:10:01: IP: s=13.1.1.1
(Serial3/0), d=10.10.10.2 (Virtual-Access1), g=10.10.10.2, len 100, forward 00:10:01: ICMP
type=0, code=0 00:10:01: IP: s=13.1.1.1 (Serial3/0), d=10.10.10.2 (Virtual-Access1),
g=10.10.10.2, len 100, forward 00:10:01: ICMP type=0, code=0
```

Laten we nu snel overschakelen op de andere interface verwijderen (op router_122). Dit betekent dat alle pakketten over deze twee interfaces nu pakketgeschakeld zijn (wat een vereiste voor **debug ip packet**):

```
router_122(config)#interface serial 3/0
router_122(config-if)#no ip route-cache
router_122(config-if)#end
```

```
router_122#
00:11:57:  IP: s=10.10.10.2 (Virtual-Access1), d=13.1.1.1
(Serial3/0), g=172.16.1.6, len 100, forward
00:11:57:  ICMP type=8, code=0
! -- ICMP packet (echo) from 10.10.10.2 to 13.1.1.1 00:11:57: IP: s=13.1.1.1 (Serial3/0),
d=10.10.10.2 (Virtual-Access1),
g=10.10.10.2, len 100, forward
00:11:57:  ICMP type=0, code=0
! -- ICMP return packet (echo-reply) from 13.1.1.1 to 10.10.10.2 00:11:57: IP: s=10.10.10.2
(Virtual-Access1), d=13.1.1.1 (Serial3/0), g=172.16.1.6, len 100, forward 00:11:57: ICMP type=8,
code=0 00:11:57: IP: s=13.1.1.1 (Serial3/0), d=10.10.10.2 (Virtual-Access1), g=10.10.10.2, len
100, forward 00:11:57: ICMP type=0, code=0
```

Merk op dat de debug ip-pakketuitvoer geen pakketten toont die niet voldoen aan de toegangs-lijst criteria. Raadpleeg voor meer informatie over deze procedure de opdrachten Ping en Traceroute begrijpen.

Zie Vastlegging standaard IP-toegangslijsten voor meer informatie over het maken van toegangslijsten.

voorwaardelijk veroorzaakte aftappen

Wanneer de voorwaardelijk teweeggebrachte Debugging optie wordt geactiveerd, genereert de router de zuiverende berichten voor pakketten die de router op een gespecificeerde interface ingaan of verlaten; de router genereert geen debugoutput voor pakketten die binnenkomen of door een andere interface vertrekken.

Kijk naar een eenvoudige implementatie van voorwaardelijke deposito's. Neem dit scenario in overweging: de router die hieronder (trabol) wordt getoond heeft twee interfaces (seriële 0 en seriële 3), beide HDLC-insluiting uitvoeren.

U kunt het normale **debug serial interface** opdracht om de op alle interfaces ontvangen toetsen van de HDLC te observeren. U kunt de keepalives op beide interfaces waarnemen.

```
traxbol#debug serial interface
Serial network interface debugging is on
traxbol#
*Mar  8 09:42:34.851: Serial0: HDLC myseq 28, mineseen 28*, yourseen 41, line up
! -- HDLC keeplaive on interface Serial 0 *Mar 8 09:42:34.855: Serial3: HDLC myseq 26, mineseen
26*, yourseen 27, line up
! -- HDLC keeplaive on interface Serial 3 *Mar 8 09:42:44.851: Serial0: HDLC myseq 29, mineseen
29*, yourseen 42, line up *Mar 8 09:42:44.855: Serial3: HDLC myseq 27, mineseen 27*, yourseen
28, line up
```

Schakel voorwaardelijke media voor interface-serienummer 3 in. Dit betekent dat alleen versies voor interface-seriële 3 worden weergegeven. Gebruik het **debug interface <*interface_type interface_number*>** uit.

```
traxbol#debug interface serial 3
Condition 1 set
```

Gebruik het **show debug condition** opdracht om te controleren of het voorwaardelijke debug actief is. Merk op dat een voorwaarde voor interface-serie 3 actief is.

```
traxbol#show debug condition
```

```
Condition 1: interface Se3 (1 flags triggered)
Flags: Se3
traxbol#
```

Merk op dat nu alleen de apparaten voor interface-seriële 3 worden weergegeven

```
*Mar  8 09:43:04.855: Serial3: HDLC myseq 29, mineseen 29*, yourseen 30, line up
*Mar  8 09:43:14.855: Serial3: HDLC myseq 30, mineseen 30*, yourseen 31, line up
```

Gebruik het **undebug interface <*interface_type interface_number*>** commando om het voorwaardelijke debug te verwijderen. Het wordt aanbevolen om de debugs uit te schakelen (bijvoorbeeld door undebug all te gebruiken) voordat u de voorwaardelijke trigger verwijdert. Dit is om een stortvloed van debug-uitgangen te voorkomen wanneer de conditie wordt verwijderd.

```
traxbol#undebug interface serial 3
This condition is the last interface condition set.
Removing all conditions may cause a flood of debugging
messages to result, unless specific debugging flags
are first removed.
Proceed with removal? [yes/no]: y
Condition 1 has been removed
traxbol
```

U kunt nu observeren dat debug voor zowel interface-serie 0 als seriële serie 3 wordt weergegeven.

```
*Mar  8 09:43:34.927: Serial3: HDLC myseq 32, mineseen 32*, yourseen 33, line up
*Mar  8 09:43:44.923: Serial0: HDLC myseq 35, mineseen 35*, yourseen 48, line up
```

> **Waarschuwing**: Sommige zuiveringsoperaties zijn door zichzelf voorwaardelijk. Een voorbeeld is het zuiveren van ATM. Met het zuiveren van ATM zou u expliciet de interface moeten specificeren waarvoor diepgang zou moeten worden geactiveerd in plaats van het toelaten van diepgang op alle ATM interfaces en het specificeren van een voorwaarde.

Deze sectie toont de juiste manier om het pakket van ATM te beperken zuiveren tot één subinterface:

```
arielle-nrp2#debug atm packet interface atm 0/0/0.1
!--- Note that you explicitly specify the sub-interface to be used for debugging ATM packets
debugging is on Displaying packets on interface ATM0/0/0.1 only
arielle-nrp2#
*Dec 21 10:16:51.891: ATM0/0/0.1(O):
VCD:0x1 VPI:0x1 VCI:0x21 DM:0x100 SAP:AAAA CTL:03 OUI:0080C2 TYPE:0007
Length:0x278
*Dec 21 10:16:51.891: 0000 FFFF FFFF FFFF 0010 7BB9 BDC4 0800 4500 025C 01FE
0000 FF11 61C8 0A30
*Dec 21 10:16:51.891: 4B9B FFFF FFFF 0044 0043 0248 0000 0101 0600 0015 23B7
0000 8000 0000 0000
*Dec 21 10:16:51.891: 0000 0000 0000 0000 0000 0000 0010 7BB9 BDC3 0000 0000
0000 0000 0000 0000
*Dec 21 10:16:51.891: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000
*Dec 21 10:16:51.891: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000
*Dec 21 10:16:51.891: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000
```

```
*Dec 21 10:16:51.891: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000
*Dec 21 10:16:51.895: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000
*Dec 21 10:16:51.895:
arielle-nrp2#
```

Als u probeert dit in te schakelen **atm debugging** op alle interfaces (met een toegepaste voorwaarde) kan de router hangen als deze een groot aantal ATM subinterfaces heeft. Een voorbeeld van de incorrecte methode voor het foutoptreden van ATM wordt weergegeven.

In dit geval kun je zien dat een voorwaarde wordt toegepast, maar je ziet ook dat dit geen effect heeft. U kunt het pakket nog steeds vanuit de andere interface zien. In dit lab scenario heb je slechts twee interfaces en heel weinig verkeer. Als het aantal interfaces hoog is, debug uitvoer voor alle interfaces extreem hoog is en kan het de router veroorzaken om op te hangen.

```
arielle-nrp2#show debugging condition
Condition 1: interface AT0/0/0.1 (1 flags triggered)
Flags: AT0/0/0.1
! -- A condition for a specific interface. arielle-nrp2#debug atm packet
ATM packets debugging is on
Displaying all ATM packets
arielle-nrp2#
*Dec 21 10:22:06.727: ATM0/0/0.2(O):
! -- You see debugs from interface ATM0/0/0/.2, even though the condition ! -- specified ONLY
AT0/0/0.1 VCD:0x2 VPI:0x5 VCI:0x37 DM:0x100 SAP:AAAA CTL:03 OUI:0080C2 TYPE:000E Length:0x2F
*Dec 21 10:22:06.727: 0000 0000 0180 0000 107B B9BD C400 0000 0080 0000 107B B9BD C480 0800 0014
*Dec 21 10:22:06.727: 0002 000F 0000 *Dec 21 10:22:06.727: un a *Dec 21 10:22:08.727:
ATM0/0/0.2(O): VCD:0x2 VPI:0x5 VCI:0x37 DM:0x100 SAP:AAAA CTL:03 OUI:0080C2 TYPE:000E
Length:0x2F *Dec 21 10:22:08.727: 0000 0000 0180 0000 107B B9BD C400 0000 0080 0000 107B B9BD
C480 0800 0014 *Dec 21 10:22:08.727: 0002 000F 0000 *Dec 21 10:22:08.727: ll *Dec 21
10:22:10.727: ATM0/0/0.2(O): VCD:0x2 VPI:0x5 VCI:0x37 DM:0x100 SAP:AAAA CTL:03 OUI:0080C2
TYPE:000E Length:0x2F *Dec 21 10:22:10.727: 0000 0000 0080 0000 107B B9BD C400 0000 0080 0000
107B B9BD C480 0800 0014 *Dec 21 10:22:10.727: 0002 000F 0000 *Dec 21 10:22:10.727: *Dec 21
10:22:12.727: ATM0/0/0.2(O): VCD:0x2 VPI:0x5 VCI:0x37 DM:0x100 SAP:AAAA CTL:03 OUI:0080C2
TYPE:000E Length:0x2F *Dec 21 10:22:12.727: 0000 0000 0080 0000 107B B9BD C400 0000 0080 0000
107B B9BD C480 0800 0014 *Dec 21 10:22:12.727: 0002 000F 0000 *Dec 21 10:22:12.727: *Dec 21
10:22:13.931: ATM0/0/0.1(O):
!--- You also see debugs for interface ATM0/0/0.1 as you wanted. VCD:0x1 VPI:0x1 VCI:0x21
DM:0x100 SAP:AAAA CTL:03 OUI:0080C2 TYPE:0007 Length:0x278 *Dec 21 10:22:13.931: 0000 FFFF FFFF
FFFF 0010 7BB9 BDC4 0800 4500 025C 027F 0000 FF11 6147 0A30 *Dec 21 10:22:13.931: 4B9B FFFF FFFF
0044 0043 0248 0000 0101 0600 001A 4481 0000 8000 0000 0000 *Dec 21 10:22:13.931: 0000 0000 0000
0000 0000 0000 0010 7BB9 BDC3 0000 0000 0000 0000 0000 0000 *Dec 21 10:22:13.931: 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 *Dec 21 10:22:13.931: 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 *Dec 21 10:22:13.931: 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 *Dec 21 10:22:13.931: 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 *Dec 21 10:22:13.935: 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
```

Gerelateerde informatie

- Ondersteuning van inbel- en toegangstechnologie
- Technische ondersteuning en documentatie – Cisco Systems