

如何為ACI使用FCS和CRC故障排除指令碼

目錄

[簡介](#)

[手動運行指令碼的前提條件](#)

[從容器運行指令碼的先決條件](#)

[執行指令碼的步驟](#)

簡介

ACI遵循直通交換，這意味著資料包在計算CRC之前已經轉發。這些資料包通常作為輸出錯誤進行儲存轉發出去。由於ACI不會丟棄這些資料包，因此同一資料包會遍歷該資料包，並且路徑上的儲存CRC計數器會遞增。這並不意味著看到CRC的所有介面都有故障。因此，需要正確的分類來隔離有問題的埠/SFP/光纖。現在通過Python指令碼自動執行分類過程，從而更輕鬆地進行故障排除並避免手動任務。本文檔的作用是解釋如何使用自動化指令碼（請參閱附件）。

手動運行指令碼的前提條件

執行指令碼的客戶端電腦需要滿足以下要求

- a. 應安裝Python3
- b. 對ACI域的網路訪問
- c. 要安裝的ACI_CRC_requirements.txt (附加)。此檔案位於此[處](#)。

將檔案(ACI_CRC_requirements.txt)下載到客戶端電腦

開啟終端並運行命令 — pip3 install -r ACI_CRC_requirements.txt

```
ABCD-M-G24X:downloads abcd$ pip3 install -r ACI_CRC_requirements.txt
```

```
Collecting bcrypt==3.2.0 (from -r ACI_CRC_requirements.txt (line 1))
```

```
Downloading
```

```
https://files.pythonhosted.org/packages/bf/6a/0afb1e04aebd4c3ceae630a87a55fbfbbd94dea4eaf01e53d36743c85f02/bcrypt-3.2.0-cp36-abi3-macosx\_10\_9\_x86\_64.whl
```

```
Collecting cffi==1.14.6 (from -r ACI_CRC_requirements.txt (line 2))
```

```
Downloading
```

```
https://files.pythonhosted.org/packages/ca/e1/015e2ae23230d9de8597e9ad8c0b81d5ac181f08f2e6e75774b7f5301677/cffi-1.14.6-cp38-cp38-macosx\_10\_9\_x86\_64.whl (176kB)
```

```
|| 184kB 1.4MB/s
```

```
**snip**
```

```
Successfully installed DateTime-4.3 Pillow-8.3.2 bcrypt-3.2.0 cffi-1.14.6 cryptography-3.4.8  
cyclor-0.10.0 kiwisolver-1.3.2
```

```
matplotlib-3.4.3 numpy-1.21.2 pandas-1.3.2 paramiko-2.7.2 pyparsing-2.4.7 python-dateutil-2.8.2  
pytz-2021.1 six-1.16.0
```

```
stdiomask-0.0.5 tabulate-0.8.9 termcolor-1.1.0 zope.interface-5.4.0
```

從容器運行指令碼的先決條件

4.成功執行第一個指令碼後，會將原始資料檔案儲存在步驟2中使用者指定的位置。

如以下示例所示進行驗證。

```
ABCD-M-G24X:FCS_Checker kbosu$ pwd
/Users/abcd/Downloads/FCS_Checker
```

```
ls -l
total 16
-rw-r--r--@ 1 kbosu staff 1419 Sep 27 11:28 CRC_FCS_20210927_1128.txt
-rw-r--r--@ 1 kbosu staff 1419 Sep 27 11:33 CRC_FCS_20210927_1133.txt
ABCD-M-G24X:FCS_Checker abcd$
```

5.現在是執行第二個指令碼(ACI_CRC_Parser.py)的時候了。

Script-2將使用由script-1建立的那些檔案，並且進一步工作。

請輸入給定群集中的某個APIC的OOB IP地址及其憑據。

此外，輸入在執行第一個指令碼時在步驟2中輸入的相同檔案位置。

```
ABCD-M-G24X:downloads abcd$ python3 ACI_CRC_Parser.py
```

```
Enter the IP address or DNS Name of APIC: 10.197.204.184
```

```
Enter the username: admin
```

```
Enter the password: *****
```

```
Trying to connect to APIC
```

```
Connection established to the APIC
```

```
Please enter the folder where files are stored
```

```
Please make sure we have at least two files exists in the directory where you have saved data
```

```
Enter the absolute path of the folder where the files are stored:/Users/abcd/Downloads/FCS_Checker/
```

```
You have CRC and FCS for the below date range
```

```
1.2021-09-27
```

```
Fetching first and last file of the same date 20210927
```

```
CRC_FCS_20210927_1128.txt
```

```
CRC_FCS_20210927_1133.txt
```

```
The script is executing.....
```

```
The script execution has completed
```

6. Script-2將以表格格式列印資料，如下例所示。

首先，它將在使用者指定的時間間隔內列出具有非零CRC和FCS錯誤的節點介面，以及其CRC/FCS計數器的差異。使用LLDP，該指令碼還將確定與給定介面連線的鄰居裝置，最重要的是，它將從交換矩陣的觀點指出哪個節點/介面是錯誤源，以及哪些節點介面剛剛看到由於Stomp導致的CRC。

從FCS故障排除的角度來看，應重點討論以「紅色」突出並標籤為「本地」的故障排除。

這可能是介面，其中錯誤/損壞的封包正從進入交換矩陣，並導致交換矩陣中的CRC泛洪。

POD_ID	NODE_ID	NODE_NAME	NODE_ROLE	INTERFACE	20210927_1128	20210927_1133	20210927_1128	20210927_1133	NEIGHBOR	ERROR SOURCE
					CRC	CRC Diff	FCS	FCS Diff		
1	302	bgl-aci06-t2-leaf2	leaf	eth1/44	5002806823759	127841888	5002806823759	127841888	No LLDP /CDP neighbours found please check physically where this interface connects	Local
1	101	bgl-aci06-spine1	spine	eth1/1	2981200154	132103050	0	0	System:bgl-aci06-t1-leaf1.cisco.com,Interface:Eth1/49 Stomp	
1	101	bgl-aci06-spine1	spine	eth1/2	968286	0	0	0	Historic	
1	201	bgl-aci06-t1-leaf1	leaf	eth1/1	12	0	0	0	Historic	
1	201	bgl-aci06-t1-leaf1	leaf	eth1/51	4999243774529	0	0	0	Historic	
1	201	bgl-aci06-t1-leaf1	leaf	eth1/52	5002807353809	127841212	0	0	System:bgl-aci06-t2-leaf2.cisco.com,Interface:Eth1/49 Stomp	
1	202	bgl-aci06-t1-leaf2	leaf	eth1/51	968286	0	0	0	Historic	
1	301	bgl-aci06-t2-leaf1	leaf	eth1/44	4999245287405	0	4999245287405	0	Historic	
1	301	bgl-aci06-t2-leaf1	leaf	eth1/49	4999823953891	0	0	0	Historic	
1	302	bgl-aci06-t2-leaf2	leaf	eth1/49	4999243774529	0	0	0	Historic	

7.此外，該指令碼將向使用者提供以下選項，以便排序和檢視由指令碼1和2收集的粒度資料。

使用者可以選擇數字1-3之間的選項作為輸入。請參見下面的示例。

- 1.Sort the data further
- 2.View the granular data of an interface
- 3.Exit

Input the number:

在下面的示例中，我們選擇了選項2，它幫助我們檢視任何給定介面的精細資料。

該指令碼將提示使用者輸入上面列印表格中各自的POD編號、節點ID和介面ID（步驟6）。

在本例中，我們使用1-302-eth1/44，其中POD ID為1，節點ID為302，介面ID為eth1/44。這是介面其中指令碼報告了本地FCS，如步驟6所示。

Input the number:2

Enter an interface for which you need granular data(POD_ID-NODE_ID-INTERFACE Example:1-101-eth1/5): **1-302-eth1/44**

You have CRC and FCS data in the below date range
1.2021-09-27

Enter the date for which you need granular data(any number from the above list range(1-1)):

在我們的示例中，我們每天只收集幾分鐘的資料，因此，我們只看到一個日期是9月27日的選項。

因此，我們的輸入是「1」。

Enter the date for which you need granular data(any number from the above list range(1-1)): 1

Time	CRC	FCS
11:28	5002806823759	5002806823759
11:33	5002934665647	5002934665647

Do you want to continue viewing the granular data(0/1), 1=yes, 0=no:0

Please select any number below to sort the data further or to view granular data of an interface

- 1.Sort the data further
- 2.View the granular data of an interface
- 3.Exit

Input the number:3

ABCD-M-G24X:downloads abcd\$