

Nexusスイッチの巡回冗長検査(CRC)エラーについて

内容

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[背景説明](#)

[該当ハードウェア](#)

[CRCの定義](#)

[CRCエラーの定義](#)

[CRCエラーの一般的な症状](#)

[Windowsホストでエラーを受信](#)

[LinuxホストでのRXエラー](#)

[ネットワークデバイスのCRCエラー](#)

[ストアアンドフォワードネットワークデバイスでの入力エラー](#)

[カットスルーネットワークデバイスでの入出力エラー](#)

[CRCエラーのトレースと切り分け](#)

[CRCエラーの根本原因](#)

[CRCエラーの解決](#)

[関連情報](#)

概要

このドキュメントでは、インターフェイスカウンタとCisco Nexusスイッチの統計情報で見られる巡回冗長検査(CRC)エラーに関する詳細について説明します。

前提条件

要件

イーサネットスイッチングとCisco NX-OSコマンドラインインターフェイス(CLI)の基本を理解しておくことをお勧めします。詳細については、次の該当ドキュメントのいずれかを参照してください。

- [Cisco Nexus 9000 NX-OS Fundamentalsコンフィギュレーションガイド、リリース10.2\(x\)](#)
- 『[Cisco Nexus 9000 Series NX-OS Fundamentals Configuration Guide, Release 9.3\(x\)](#)』
- 『[Cisco Nexus 9000 Series NX-OS Fundamentals Configuration Guide, Release 9.2\(x\)](#)』
- 『[Cisco Nexus 9000 Series NX-OS Fundamentals Configuration Guide, Release 7.x](#)』
- [イーサネットのトラブルシューティング](#)

使用するコンポーネント

このドキュメントの情報は、次のソフトウェアとハードウェアのバージョンに基づいています。

- NX-OSソフトウェアリリース9.3(8)以降のNexus 9000シリーズスイッチ
- NX-OSソフトウェアリリース9.3(8)以降のNexus 3000シリーズスイッチ

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されたものです。このドキュメントで使用するすべてのデバイスは、初期（デフォルト）設定の状態から起動しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、初期（デフォルト）設定の状態から起動しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

背景説明

このドキュメントでは、Cisco Nexusシリーズスイッチのインターフェイスカウンタで見られる巡回冗長検査(CRC)エラーに関する詳細について説明します。このドキュメントでは、イーサネットフレームのFrame Check Sequence (FCS ; フレームチェックシーケンス) フィールドでのCRCの使用方法、NexusスイッチでのCRCエラーの発生方法、ストアアンドフォワードスイッチングとカットスルースイッチングのシナリオでのCRCエラーの発生原因、およびCRCエラーの解決方法について説明します。

該当ハードウェア

このドキュメントの情報は、すべてのCisco Nexusシリーズスイッチに適用されます。このドキュメントの情報の一部は、Cisco Catalystルータやスイッチなど、他のシスコルーティングおよびスイッチングプラットフォームにも適用できます。

CRCの定義

CRCは、コンピュータおよびストレージネットワークで一般的に使用されるエラー検出メカニズムで、伝送中に変更または破損したデータを識別します。ネットワークに接続されたデバイスがデータを送信する必要がある場合、デバイスは固定長のデータに対して計算アルゴリズムをします。この固定長の番号はCRC値と呼ばれますが、一般的に、短い場合はCRCと呼ばれることがよくあります。このCRC値はデータに付加され、ネットワークを介して別のデバイスに送信されます。このリモートデバイスは、データに対して同じ巡回符号アルゴリズムを実行し、結果の値とデータに付加されたCRCを比較します。両方の値が一致する場合、リモートデバイスはデータが破損することなくネットワーク経由で送信されたと見なします。値が一致しない場合、リモートデバイスはネットワークを介した伝送中にデータが破損したと見なします。この破損データは信頼できず、破棄されます。

CRCは、イーサネット（有線および無線の両方のバリエーション）、トークンリング、非同期転送モード(ATM)、フレームリレーなどの複数のコンピュータネットワークテクノロジーのエラー検出に使用されます。イーサネットフレームの末尾（フレームのペイロード直後）に32ビットのFrame Check Sequence(FCS)フィールドがあります挿入されました。

たとえば、Host-AとHost-Bという2台のホストがネットワークインターフェイスカード(NIC)を介して互いに直接接続されているシナリオを考えます。Host-Aは、ネットワークを介してHost-Bに「これは例です」という文を送信する必要があります。Host-Aは、Host-Bを宛先とするイーサネットフレームを「This is an example」というペイロードで細工し、フレームのCRC値が0xABCDの16進数値であると計算します。Host-Aは、イーサネットフレームのFCSフィールドに0xABCDのCRC値を挿入し、Host-AのNICからHost-Bにイーサネットフレームを送信します。

Host-Bはこのフレームを受信すると、Host-Aとまったく同じアルゴリズムを使用してフレームのCRC値を計算します。Host-Bは、フレームのCRC値が0xABCDの16進数値であることを計算します。これは、フレームがHost-Bに送信されている間にイーサネットフレームが破損していないことをHost-Bに示します。

CRCエラーの定義

CRCエラーは、デバイス(ネットワークデバイスまたはネットワークに接続されたホスト)が、フレームのFCSフィールドにCRC値を持つイーサネットフレームを受信し、フレームのデバイスによって計算されたCRC値と一致しない場合に発生します。

この概念は、例を通じて最も効果的に示されています。Host-AとHost-Bという2つのホストが、ネットワークインターフェイスカード(NIC)を介して相互に直接接続されているシナリオを考えます。Host-Aは、ネットワークを介してHost-Bに「これは例です」という文を送信する必要があります。Host-Aは、「This is an example」というペイロードを持つHost-B宛てのイーサネットフレームを細工し、フレームのCRC値が16進数値0xABCDであることを計算します。Host-Aは、イーサネットフレームのFCSフィールドに0xABCDのCRC値を挿入し、Host-AのNICからHost-Bにイーサネットフレームを送信します。

ただし、Host-AとHost-Bを接続する物理メディアが破損すると、フレーム内の文が目的のペイロード「This is an example」ではなく「This was an example」に変わるように、フレームの内容が破損します。

Host-Bはこのフレームを受信すると、破損したペイロードを含むフレームのCRC値を計算します。Host-Bは、フレームのCRC値が0xDEADの16進数値であることを計算します。これは、イーサネットフレームのFCSフィールド内の0xABCD CRC値とは異なります。このCRC値の違いは、Host-Bに送信できませんこのイーサネットフレームの内容をドロップします。Host-Bは通常、ネットワークインターフェイスカード(NIC)の「input errors」、「CRC errors」、「RX errors」などのエラーカウンタも増加します。

CRCエラーの一般的な症状

CRCエラーは通常、次の2つの方法のいずれかで発生します。

1. ネットワークに接続されたデバイスのインターフェイスのエラーカウンタが増加またはゼロ以外の場合。
2. ネットワークに接続されたデバイスが破損したフレームをドロップしたために、ネットワークを通過するトラフィックのパケット/フレーム損失。

これらのエラーは、使用しているデバイスによって若干異なる方法で発生します。これらのサブセクションでは、デバイスのタイプごとに詳細を説明します。

Windowsホストでエラーを受信

WindowsホストのCRCエラーは、通常、コマンドプロンプトからのnetstat -eコマンドの出力にゼロ以外のReceived Errorsカウンタが表示されます。Windowsホストのコマンドプロンプトからのゼロ以外のReceived Errorsカウンタの例はです。

```
>netstat -e
Interface Statistics

                Received                Sent
Bytes           1116139893           3374201234
Unicast packets 101276400             49751195
Non-unicast packets 0                     0
Discards        0                     0
Errors         47294                0
Unknown protocols 0
```

netstat -eコマンドによって報告される受信エラーの数が正確になるためには、NICとそれぞれのドライバがNICによって受信されるCRCエラーのアカウントリングをサポートしている必要があります。最近のほとんどのNICとそれぞれのドライバは、NICが受信するCRCエラーの正確なアカウントリングをサポートしています。

LinuxホストでのRXエラー

Linuxホスト上のCRCエラーは通常、ifconfigコマンドの出力にゼロ以外の「RXエラー」カウンタとして表示されます。Linuxホストからのゼロ以外のRXエラーカウンタの例を次に示します。

```
$ ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.0.2.10 netmask 255.255.255.128 broadcast 192.0.2.255
    inet6 fe80::10 prefixlen 64 scopeid 0x20<link>
    ether 08:62:66:be:48:9b txqueuelen 1000 (Ethernet)
    RX packets 591511682 bytes 214790684016 (200.0 GiB)
    RX errors 478920 dropped 0 overruns 0 frame 0
    TX packets 85495109 bytes 288004112030 (268.2 GiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Linuxホスト上のCRCエラーは、ip -s link showコマンドの出力にゼロ以外の「RX errors」カウンタとして表示されることもあります。Linuxホストからのゼロ以外のRXエラーカウンタの例を次に示します。

```
$ ip -s link show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode DEFAULT group
default qlen 1000
    link/ether 08:62:66:84:8f:6d brd ff:ff:ff:ff:ff:ff
    RX: bytes  packets  errors  dropped overrun mcast
    32246366102 444908978 478920    647      0        419445867
    TX: bytes  packets  errors  dropped carrier collsns
    3352693923 30185715 0        0        0        0
    altname enp11s0
```

ifconfigコマンドまたはip -s link showコマンドで報告されるRXエラーの数を正確にするために、NICとそれぞれのドライバは、NICが受信したCRCエラーのアカウントリングをサポートしている必要があります。最近のほとんどのNICとそれぞれのドライバは、NICが受信するCRCエラーの正確なアカウントリングをサポートしています。

ネットワークデバイスのCRCエラー

ネットワークデバイスは、ストアアンドフォワードフォーディングモードとカットスルーフォーディングモードの2つのフォーディングモードのいずれかで動作します。ネットワークデバイスが受信したCRCエラーを処理する方法は、転送モードによって異なります。ここでは、各転送モードの具体的な動作について説明します。

ストアアンドフォワードネットワークデバイスでの入力エラー

ストアアンドフォワードフォーディングモードで動作するネットワークデバイスがフレームを受信すると、フレームのCRC値を検証し、フレームの転送決定を行い、インターフェイスからフレームを送信する（「転送」）前に、フレーム全体をバッファリングします。したがって、ストアアンドフォワードフォーディングモードで動作するネットワークデバイスが、特定のインターフェイスで不正なCRC値を持つ破損したフレームを受信すると、そのフレームを廃棄し、インターフェイスの「Input Errors」カウンタを増加させます。

つまり、破損したイーサネットフレームは、ストアアンドフォワードフォーディングモードで動作するネットワークデバイスによって転送されません。これらは入力時に廃棄されます。

Cisco Nexus 7000および7700シリーズスイッチは、ストアアンドフォワードフォーディングモードで動作します。Nexus 7000または7700シリーズスイッチからの非ゼロ入力エラーカウンタと非ゼロCRC/FCSカウンタの例を次に示します。

```
switch# show interface
<snip>
Ethernet1/1 is up
RX
 241052345 unicast packets  5236252 multicast packets  5 broadcast packets
 245794858 input packets  17901276787 bytes
 0 jumbo packets  0 storm suppression packets
 0 runts  0 giants  579204 CRC/FCS  0 no buffer
 579204 input error  0 short frame  0 overrun  0 underrun  0 ignored
 0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
 0 input with dribble  0 input discard
 0 Rx pause
```

また、CRCエラーは、**show interface counters errors**の出力でゼロ以外の「FCS-Err」カウンタとして現在する可能性があります。このコマンドの出力の「Rcv-Err」カウンタにもゼロ以外の値が設定されます。これは、インターフェイスで受信されたすべての入力エラー（CRCまたはその他のエラー）の合計です。次に例を示します。

```
switch# show interface counters errors
<snip>
-----
Port                Align-Err    FCS-Err    Xmit-Err    Rcv-Err    UnderSize  OutDiscards
-----
Eth1/1                0          579204         0          579204         0           0
```

カットスルーネットワークデバイスでの入出力エラー

カットスルーフォーディングモードで動作するネットワークデバイスがフレームの受信を開始すると、ネットワークデバイスはフレームのヘッダーに転送決定を行い、有効な転送決定を行うために十分なフレームを受信するとすぐにインターフェイスからフレームの送信を開始します。フレームとパケットヘッダーはフレームの先頭にあるため、通常、フレームのペイロードが受信される前にこの転送決定が行われます。

イーサネットフレームのFCSフィールドは、フレームのペイロードの直後のフレームの末尾にあ

ります。したがって、カットスルーフォワーディングモードで動作するネットワークデバイスは、フレームのCRCを計算できる時間までに、別のインターフェイスからフレームの送信を開始しています。ネットワークデバイスによってフレームに対して計算されたCRCが、FCSフィールドに存在するCRC値と一致しない場合、ネットワークデバイスは破損したフレームをネットワークに転送したことを意味します。この場合、ネットワークデバイスは2つのカウンタを増加します。

1. 破損したフレームが最初に受信されたインターフェイスの「Input Errors」カウンタ。
2. 破損したフレームが送信されたすべてのインターフェイスの「Output Errors」カウンタ。ユニキャストトラフィックの場合、通常は1つのインターフェイスになります。ただし、ブロードキャスト、マルチキャスト、または不明なユニキャストトラフィックの場合は、1つ以上のインターフェイスを指定できます。

この例を次に示します。show interfaceコマンドの出力は、ネットワークデバイスのEthernet1/1で複数の破損フレームが受信され、ネットワークデバイスのカットスルーフォワーディングモードによりEthernet1/2から送信されたことを示します。

```
switch# show interface
<snip>
Ethernet1/1 is up
RX
 46739903 unicast packets 29596632 multicast packets 0 broadcast packets
 76336535 input packets 6743810714 bytes
 15 jumbo packets 0 storm suppression bytes
 0 runts 0 giants 47294 CRC 0 no buffer
 47294 input error 0 short frame 0 overrun 0 underrun 0 ignored
 0 watchdog 0 bad etype drop 0 bad proto drop 0 if down drop
 0 input with dribble 0 input discard
 0 Rx pause

Ethernet1/2 is up
TX
 46091721 unicast packets 2852390 multicast packets 102619 broadcast packets
 49046730 output packets 3859955290 bytes
 50230 jumbo packets
 47294 output error 0 collision 0 deferred 0 late collision
 0 lost carrier 0 no carrier 0 babble 0 output discard
 0 Tx pause
```

CRCエラーは、show interface counters errorsの出力で、入力インターフェイスではゼロ以外の「FCS-Err」カウンタ、出力インターフェイスではゼロ以外の「Xmit-Err」カウンタとして表示されることもあります。このコマンドの出力にある入力インターフェイスの「Rcv-Err」カウンタにもゼロ以外の値が設定されます。これは、インターフェイスで受信されたすべての入力エラー（CRCまたはその他）の合計です。次に例を示します。

```
switch# show interface counters errors
<snip>
-----
Port          Align-Err  FCS-Err  Xmit-Err  Rcv-Err  UnderSize  OutDiscards
-----
Eth1/1        0          47294    0          47294    0           0
Eth1/2        0           0        47294    0           0           0
```

ネットワークデバイスは、フレームのFCSフィールドのCRC値も、このフレームが破損していることをアップストリームネットワークデバイスに示す特定の方法で変更します。この動作は、CRCを「ストップ」と呼ばれます。CRCが修正される正確な方法は、プラットフォームによって異なりますが、通常は、フレームのFCSフィールドに存在する現在のCRC値を反転させる必要があります。次に例を示します。

Original CRC: 0xABCD (1010101111001101)

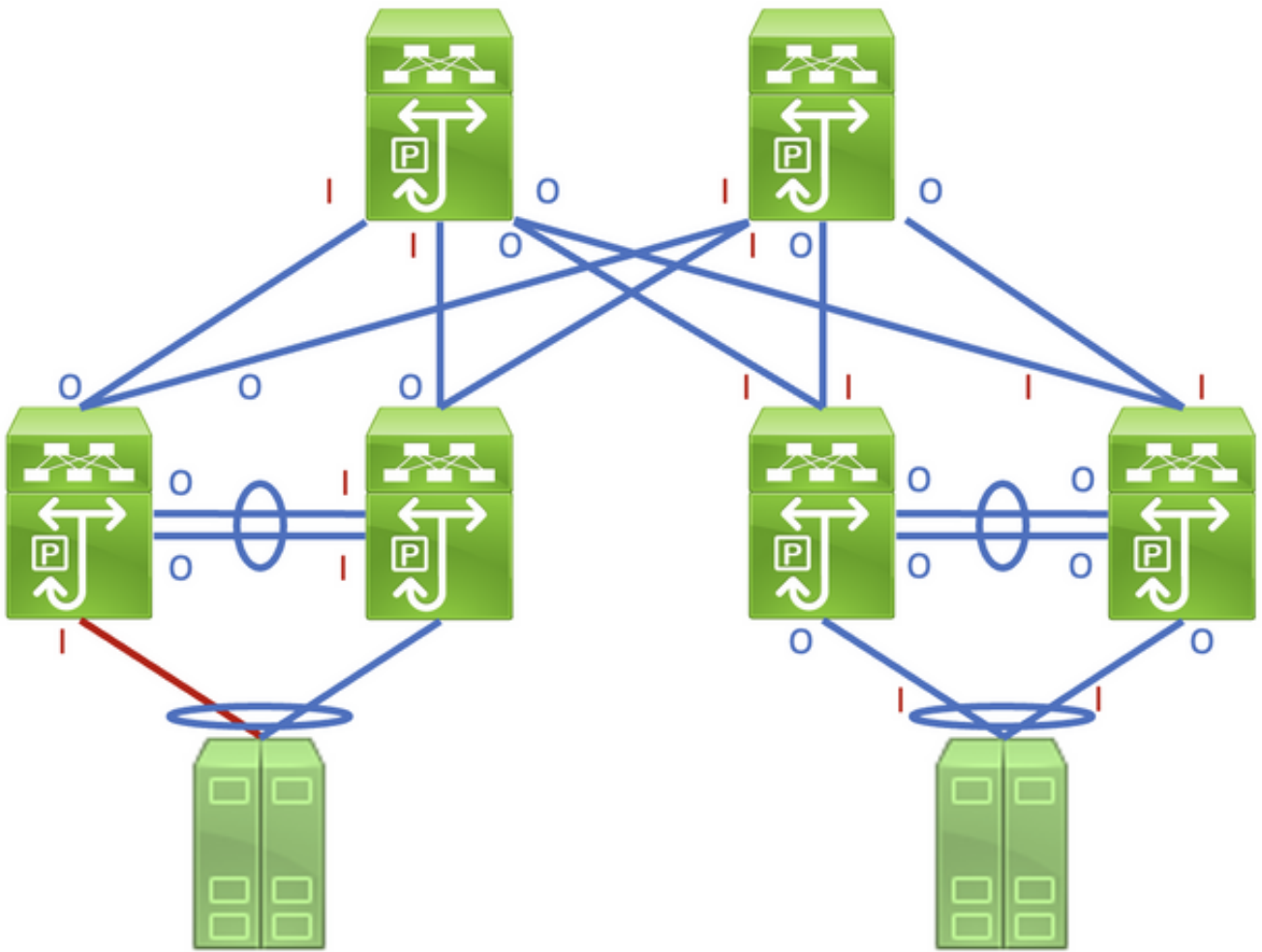
Stomped CRC: 0x5432 (0101010000110010)

この動作の結果、カットスルーフォーディングモードで動作するネットワークデバイスは、ネットワーク全体に破損したフレームを伝搬します。ネットワークがカットスルーフォーディングモードで動作する複数のネットワークデバイスで構成されている場合、破損したフレームが入力エラーと出力エラーカウンタが増加します。

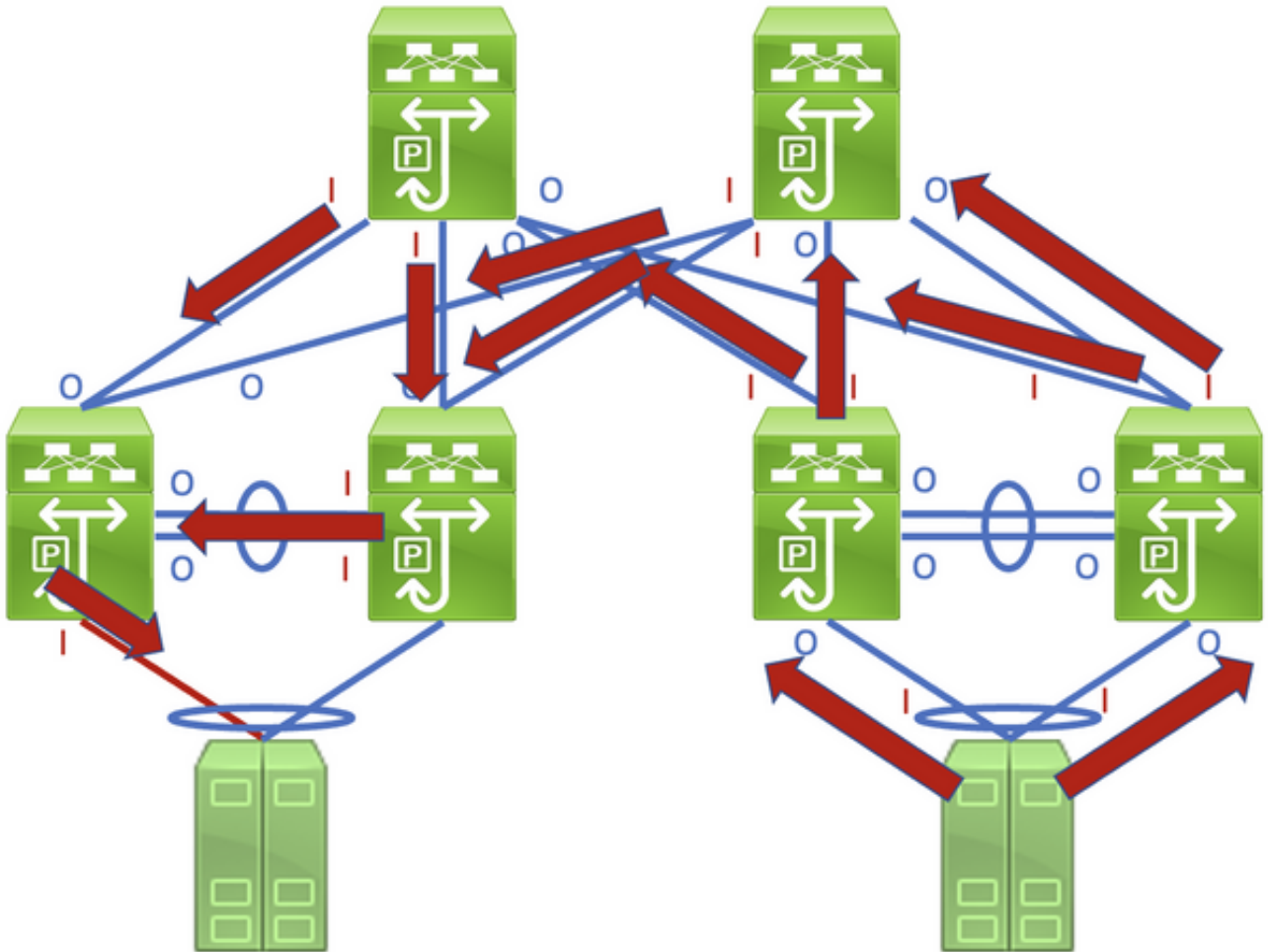
CRCエラーのトレースと切り分け

CRCエラーの根本原因を特定して解決するための最初のステップは、CRCエラーの原因を、ネットワーク内の2つのデバイス間の特定のリンクに切り分けることです。このリンクに接続された1台のデバイスのインターフェイス出力エラーカウンタはゼロまたは増加していません。一方、このリンクに接続された他のデバイスのインターフェイス入力エラーカウンタはゼロ以外または増加しています。これは、トラフィックが1つのデバイスのインターフェイスをそのまま残してリモートデバイスに送信したときに破損し、リンク上の他のデバイスの入力インターフェイスによって入力エラーとしてカウントされることを示します。

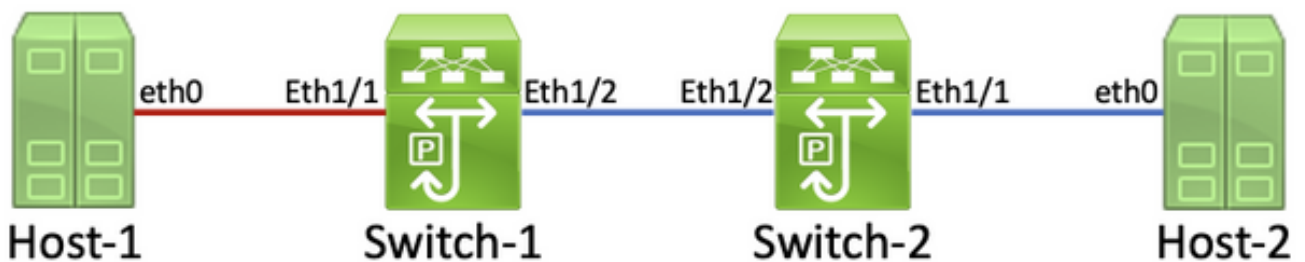
ストアアンドフォワードフォーディングモードで動作するネットワークデバイスで構成されるネットワークでこのリンクを特定することは、簡単な作業です。ただし、カットスルーフォーディングモードで動作するネットワークデバイスで構成されるネットワークでこのリンクを特定することは、多くのネットワークデバイスでゼロ以外の入力および出力エラーカウンタが使用されるため、より困難です。この現象の例は、赤で強調表示されたリンクが破損し、リンクを通過するトラフィックが破損しているトポロジで確認できます。赤色の「I」のラベルが付いたインターフェイスはゼロ以外の入力エラーが発生する可能性があるインターフェイスを示し、青色の「O」が付いたインターフェイスはゼロ以外の出力エラーが発生する可能性があるインターフェイスを示します。



障害のあるリンクを特定するには、ゼロ以外の入力および出力エラーカウンタを介してネットワーク内の「パス」破損フレームを再帰的にトレースする必要があります。ゼロ以外の入力エラーは、ネットワーク内の破損したリンクを指します。これは、次の図で示されています。



破損したリンクをトレースして特定する詳細なプロセスは、例を通じて最もよく示されています。トポロジを次に示します。



このトポロジでは、Switch-1という名前のNexusスイッチのインターフェイスEthernet1/1は、Host-1のNetwork Interface Card(NIC)eth0を介してHost-1という名前のホストに接続されます。Switch-1のインターフェイスEthernet1/2は、Switch-2のインターフェイスEthernet1/1を経由です。のスイッチ2は、ホスト2のNIC eth0を介してホスト2というホストに接続されます。

Host-1とSwitch-1のEthernet1/1インターフェイスを介したSwitch-1間のリンクが破損しており、リンクを通過するトラフィックが断続的に破損する原因になります。ただし、このリンクが破損していることはまだ分かりません。破損したフレームがネットワークに残るパスをゼロ以外の値でトレースするか、入力および出力エラーカウンタを増加させて、このネットワーク内の破損したリンクを特定する必要があります。

この例では、Host-2のNICがCRCエラーを受信していることを報告しています。

```
Host-2$ ip -s link show eth0
```

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode DEFAULT group
default qlen 1000
link/ether 00:50:56:84:8f:6d brd ff:ff:ff:ff:ff:ff
  RX: bytes  packets  errors  dropped overrun mcast
32246366102 444908978 478920    647      0      419445867
  TX: bytes  packets  errors  dropped carrier collsns
3352693923 30185715 0        0        0        0
altname enp11s0
```

Host-2のNICがインターフェイスEthernet1/1を介してSwitch-2に接続されていることがわかります。インターフェイスEthernet1/1にゼロ以外の出力エラーカウンタがあることをshow interfaceコマンドで確認できます。

```
Switch-2# show interface
```

```
<snip>
```

```
Ethernet1/1 is up
```

```
admin state is up, Dedicated Interface
```

```
  RX
```

```
30184570 unicast packets  872 multicast packets  273 broadcast packets
30185715 input packets  3352693923 bytes
0 jumbo packets  0 storm suppression bytes
0 runts  0 giants  0 CRC  0 no buffer
0 input error  0 short frame  0 overrun  0 underrun  0 ignored
0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
0 input with dribble  0 input discard
0 Rx pause
```

```
  TX
```

```
444907944 unicast packets  932 multicast packets  102 broadcast packets
444908978 output packets  32246366102 bytes
0 jumbo packets
478920 output error  0 collision  0 deferred  0 late collision
0 lost carrier  0 no carrier  0 babble  0 output discard
0 Tx pause
```

インターフェイスEthernet1/1の出力エラーカウンタがゼロでないため、ゼロ以外の入力エラーカウンタを持つスイッチ2の別のインターフェイスが存在する可能性が高くなります。show interface counters errors non-zeroコマンドを使用すると、Switch-2のインターフェイスにゼロ以外の入力エラーカウンタがあるかどうかを確認できます。

```
Switch-2# show interface counters errors non-zero
```

```
<snip>
```

```
-----
Port          Align-Err    FCS-Err    Xmit-Err    Rcv-Err    UnderSize  OutDiscards
-----
Eth1/1          0            0    478920          0            0            0
Eth1/2          0    478920          0    478920          0            0
-----
```

```
-----
Port          Single-Col  Multi-Col  Late-Col  Exces-Col  Carri-Sen    Runts
-----
```

```
-----
Port          Giants  SQETest-Err  Deferred-Tx  IntMacTx-Er  IntMacRx-Er  Symbol-Err
-----
```

```
-----
Port          InDiscards
-----
```

Switch-2のEthernet1/2にゼロ以外の入力エラーカウンタがあることがわかります。これは、Switch-2がこのインターフェイスで破損したトラフィックを受信することを示唆しています。Cisco Discovery Protocol(CDP)またはLink Local Discovery Protocol(LLDP)機能を使用して、Switch-2のEthernet1/2に接続されているデバイスを確認できます。この例をshow cdp neighborsコマンドで示します。

```
Switch-2# show cdp neighbors
```

```
<snip>
```

```
Capability Codes: R - Router, T - Trans-Bridge, B - Source-Route-Bridge  
S - Switch, H - Host, I - IGMP, r - Repeater,  
V - VoIP-Phone, D - Remotely-Managed-Device,  
s - Supports-STP-Dispute
```

```
Device-ID           Local Intrfce  Hldtme Capability  Platform          Port ID  
Switch-1(FD012345678)  
                  Eth1/2         125      R S I s      N9K-C93180YC-    Eth1/2
```

Switch-2がSwitch-1のEthernet1/2インターフェイスからEthernet1/2インターフェイスで破損したトラフィックを受信していますが、Switch-1のEthernet1/2とSwitch-2のEthernet1/2間のリンクが破損して破損しているか、またはSwitch-1が破損しています。これを確認するには、Switch-1にログインする必要があります。

show interfacesコマンドを使用すると、Switch-1のEthernet1/2インターフェイスにゼロ以外の出力エラーカウンタがあることを確認できます。

```
Switch-1# show interface
```

```
<snip>
```

```
Ethernet1/2 is up
```

```
admin state is up, Dedicated Interface
```

```
RX
```

```
30581666 unicast packets 178 multicast packets 931 broadcast packets  
30582775 input packets 3352693923 bytes  
0 jumbo packets 0 storm suppression bytes  
0 runts 0 giants 0 CRC 0 no buffer  
0 input error 0 short frame 0 overrun 0 underrun 0 ignored  
0 watchdog 0 bad etype drop 0 bad proto drop 0 if down drop  
0 input with dribble 0 input discard  
0 Rx pause
```

```
TX
```

```
454301132 unicast packets 734 multicast packets 72 broadcast packets  
454301938 output packets 32246366102 bytes  
0 jumbo packets  
478920 output error 0 collision 0 deferred 0 late collision  
0 lost carrier 0 no carrier 0 babble 0 output discard  
0 Tx pause
```

Switch-1のEthernet1/2の出力エラーカウンタが0以外であることがわかります。これは、Switch-1のEthernet1/2とSwitch-2のEthernet1/2の間のリンクが破損していないことを示しています。代わりに、Switch-1は、他のインターフェイスで受信する破損したトラフィックを転送するカッタスルースイッチです。Switch-2で以前に示したように、show interface counters errors non-zeroコマンドを使用して、Switch-1のインターフェイスにゼロ以外の入力エラーカウンタがあるかどうかを確認できます。

```
Switch-1# show interface counters errors non-zero
```

<snip>

```
-----  
Port          Align-Err    FCS-Err    Xmit-Err    Rcv-Err    UnderSize  OutDiscards  
-----  
Eth1/1                0    478920          0    478920          0          0  
Eth1/2                0          0    478920          0          0          0  
-----  
Port          Single-Col  Multi-Col   Late-Col   Exces-Col   Carri-Sen       Runts  
-----  
Port          Giants SQETest-Err Deferred-Tx IntMacTx-Er IntMacRx-Er Symbol-Err  
-----  
Port          InDiscards  
-----
```

Switch-1のEthernet1/1にゼロ以外の入力エラーカウンタがあることがわかります。これは、Switch-1がこのインターフェイスで破損したトラフィックを受信していることを示しています。このインターフェイスがHost-1のeth0 NICに接続されていることがわかります。Host-1のeth0 NICインターフェイスの統計情報を確認して、Host-1がこのインターフェイスから破損したフレームを送信するかどうかを確認できます。

```
Host-1$ ip -s link show eth0  
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode DEFAULT group  
default qlen 1000  
  link/ether 00:50:56:84:8f:6d brd ff:ff:ff:ff:ff:ff  
  RX: bytes  packets  errors dropped overrun mcast  
  73146816142 423112898 0        0        0        437368817  
  TX: bytes  packets  errors dropped carrier collsns  
  3312398924 37942624 0        0        0        0  
  altname enp11s0
```

Host-1のeth0 NIC統計情報は、ホストが破損したトラフィックを送信していないことを示しています。これは、Host-1のeth0とSwitch-1のEthernet1/1の間のリンクが破損しており、このトラフィックの破損の原因であることを示しています。このリンクでは、この破損の原因となっている障害のあるコンポーネントを特定して交換するために、さらにトラブルシューティングを行う必要があります。

CRCエラーの根本原因

CRCエラーの最も一般的な根本原因は、2つのデバイス間の物理リンクの損傷または誤動作のコンポーネントです。次に例を示します。

- 物理メディア (銅線またはファイバ) または直接接続ケーブル(DAC)の障害または破損。
- トランシーバ/光ファイバの障害または破損
- パッチパネルポートの障害または破損。
- 障害のあるネットワークデバイスハードウェア(特定のポート、ラインカード特定用途向け集積回路(ASIC)、メディアアクセスコントロール(MAC)、ファブリックモジュールなど)
- 正常に機能していないネットワークインターフェイスカードがホストに挿入されている。

また、1つ以上の誤った設定のデバイスが、ネットワーク内で誤ってCRCエラーを引き起こす可能性があります。たとえば、ネットワーク内の2つ以上のデバイス間で最大伝送ユニット(MTU)設

定の不一致が発生し、大きなパケットが誤って切り捨てられます。この設定の問題を特定して解決すると、ネットワーク内のCRCエラーも修正できます。

CRCエラーの解決

特定の動作不良コンポーネントは、除去プロセスによって識別できます。

1. 物理メディア (銅線またはファイバ) またはDACを同じタイプの正常な物理メディアに交換します。
2. あるデバイスのインターフェイスに挿入されたトランシーバを、同じモデルの正常なトランシーバに交換します。これでCRCエラーが解決しない場合は、他のデバイスのインターフェイスに挿入されているトランシーバを、同じモデルの正常なトランシーバに交換します。
3. 破損したリンクの一部としてパッチパネルが使用されている場合は、パッチパネルの正常なポートにリンクを移動します。または、可能であればパッチパネルを使用せずにリンクを接続し、潜在的な根本原因としてパッチパネルを除去します。
4. 破損したリンクを各デバイスの別の正常なポートに移動します。MAC、ASIC、またはラインカードの障害を切り分けるには、複数の異なるポートをテストする必要があります。
5. 破損したリンクにホストが関係する場合は、そのリンクをホスト上の別のNICに移動します。または、破損したリンクを正常なホストに接続して、ホストのNICの障害を切り分けます。

正常に機能していないコンポーネントがシスコ製品で (シスコネットワークデバイスやトランシーバなど)、有効なサポート契約の対象である場合は、[Cisco TACにサポートケースをオープンして、不具合のあるコンポーネントをReturn Material Authorization\(RMA\)で交換してください。](#)

関連情報

- [Nexus 9000クラウドスケールASIC CRCの識別とトレース手順](#)
- [テクニカル サポートとドキュメント – Cisco Systems](#)