

Nexus 9500-R, Nexus 3000-R: Fehlerbehebung bei Input discard

Inhalt

[Einleitung](#)

[Hintergrundinformationen](#)

[Ingress Traffic Manager \(ITM\)](#)

[Eingangs-VOQ-Puffer](#)

[Zeitplan- und Flusststeuerung](#)

[Häufige Ursachen](#)

[Anwendbare Hardware](#)

[Fehlerbehebung bei Input discard](#)

[Häufiges Szenario - Datenverkehrsfluss von 10 G zu 1 G - Konstanter Rückgang:](#)

[Schritt 1: Überprüfen Sie, welche Warteschlange auf Ihrer Input discard-Schnittstelle betroffen ist.](#)

[Schritt 2: Prüfen Sie die grafische Darstellung der für die Diagnose verwendeten Zähler von Broadcom:](#)

[Schritt 3: Finden Sie heraus, zu welchen ASICs und zu welchen Jericho Port Ihre Input discard Front-Panel-Ports gehören:](#)

[Schritt 4: Erfahren Sie, welche VOQ- und VOQ-Anschlüsse Ihr Eingangsport hat.](#)

[Schritt 5: Überprüfen Sie aus BCM-Sicht, welche Warteschlange speziell nicht leer ist. d. h. überlastet.](#)

[Schritt 6: Suchen Sie den konfigurierten Ausgangsport aus dem Wert für die nicht leere Warteschlange:](#)

[Schritt 7: Überprüfen Sie, welcher Frontblenden-Port sich im ASIC 1 befindet, und ordnen Sie ihn Jericho Port 9 entsprechend Ihrer vorherigen Suche zu.](#)

[Zusätzliche Befehle](#)

[Zusätzliche Labortests:](#)

[Schritt 1: Input discard mit mehreren konvergierten Ausgangsschnittstellen.](#)

[Schritt 2: Input discard aufgrund von SPAN.](#)

[Schritt 3: Input discard aufgrund von Traffic Hair Pin.](#)

[Schritt 4: Senden von Paketen mit einer Ziel-IP, die unbekannt ist.](#)

[Schritt 5: Input discard, während ein Zugangs-/Trunk-Port in den STP-Weiterleitungsstatus wechselt](#)

[Schritt 6: Input discard aufgrund von Eth1/9, die die Leitungsgeschwindigkeit überschreiten.](#)

Einleitung

Dieses Dokument beschreibt die Ursachen und Lösungen für input discard für den Cisco Nexus 9500-R EoR und den Nexus 3000-R ToR. Eine input discard gibt die Anzahl der Pakete an, die aufgrund einer Überlastung in der Eingabewarteschlange verworfen wurden. Diese Zahl enthält Tropfen, die durch Schwanzfallen und Weighted Random Early Detection (WRED) verursacht werden.

Wenn Sie einen Zufallsverlust, einen sporadischen oder historischen (d. h. nicht mehr

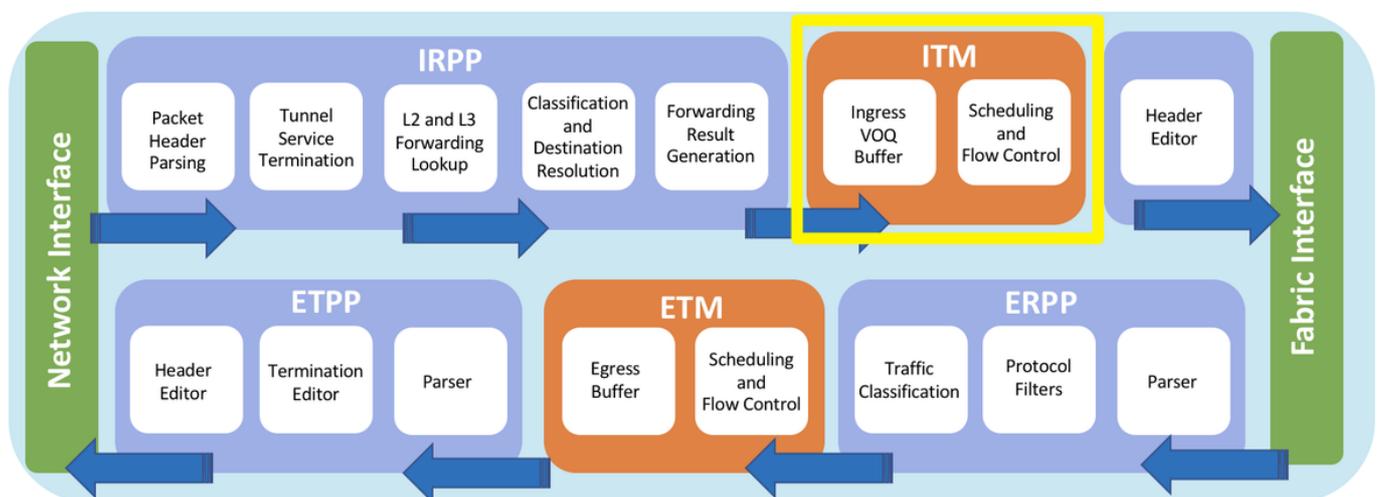
auf tretenden) Rückgang feststellen, wenden Sie sich bitte an das Cisco TAC, um weitere Informationen zu erhalten. Diese exemplarische Vorgehensweise ist hilfreich, wenn Input discard häufig erhöht werden.

Hintergrundinformationen

Die R-Serie verwendet die Eingangs-VOQ-Architektur. Die VOQ-Architektur emuliert Ausgangswarteschlangen im Eingangspuffer mit virtuellen Warteschlangen. Jeder Ausgangsport verfügt über acht Warteschlangen für Unicast-Datenverkehr und acht Warteschlangen für Multicast-Datenverkehr. Der Datenverkehr kann anhand des CoS- (Class of Service) oder DSCP-Werts (Differentiated Services Code Point) in den Paketen in Datenverkehrsklassen eingeteilt und dann in die entsprechende virtuelle Warteschlange für diese Datenverkehrsklasse aufgenommen werden.

Die R-Serie verwendet einen verteilten Kreditmechanismus, um Datenverkehr über die Fabric zu übertragen. Bevor ein Paket den VOQ verlassen soll, fordert der Eingangspuffer-Scheduler eine Gutschrift für den bestimmten Port und die Priorität im Ausgangspuffer an. Die Gutschrift wird von einem Eingangs-Credit Scheduler für den Zielport und die Priorität angefordert. Wenn Puffer-Speicherplatz verfügbar ist, gewährt der Egress-Scheduler Zugriff und sendet das Gutschrift-Grant an den Eingangspuffer-Scheduler. Wenn im Ausgangspuffer kein Pufferspeicher verfügbar ist, wird im Ausgangs-Zeitplan keine Gutschrift gewährt, und der Datenverkehr wird im VOQ gepuffert, bis die nächste Gutschrift verfügbar ist.

Unten sehen Sie die Paketweiterleitungspipeline für die -R-Plattform. In diesem Artikel konzentrieren Sie sich auf die Komponente **Ingress Traffic Manager**. Weitere Informationen zur Architektur finden Sie unter diesem [Link](#).



Ingress Traffic Manager (ITM)

Der Eingangs-Traffic-Manager (ITM) ist ein Block in der Eingangspipeline. Er führt Schritte im Zusammenhang mit dem Warteschlangendatenverkehr in VOQ durch, plant den Datenverkehr für die Übertragung über die Fabric und verwaltet Gutschriften.

Eingangs-VOQ-Puffer

Der Eingangs-VOQ-Pufferblock verwaltet sowohl den On-Chip-Puffer als auch den Off-Chip-Paket-Puffer. Beide Puffer nutzen die VOQ-Architektur, und der Datenverkehr wird anhand der

Informationen des IRPP (Ingress Receiver Packet Processor) in Warteschlangen gestellt. Insgesamt stehen 96.000 VOQs für Unicast- und Multicast-Datenverkehr zur Verfügung.

Zeitplan- und Flusststeuerung

Bevor ein Paket von der Eingangspipeline übertragen wird, muss das Paket für die Übertragung über die Fabric geplant werden. Der Eingangs-Scheduler sendet eine Kreditanfrage an den Ausgangs-Scheduler, der sich im Ausgangs-Datenverkehrsmanagement-Block befindet. Wenn der Manager für eingehenden Datenverkehr eine Gutschrift erhält, sendet er Datenverkehr an den Prozessor für eingehende Übertragungspakete. Wenn der Ausgangspuffer voll ist, wird der Datenverkehr in der dedizierten Warteschlange gepuffert, die durch den Ausgangsport und die Datenverkehrsklasse dargestellt wird.

Häufige Ursachen

Im Allgemeinen sind input discard aus den folgenden Gründen für verschiedene Nexus-Hardware sichtbar

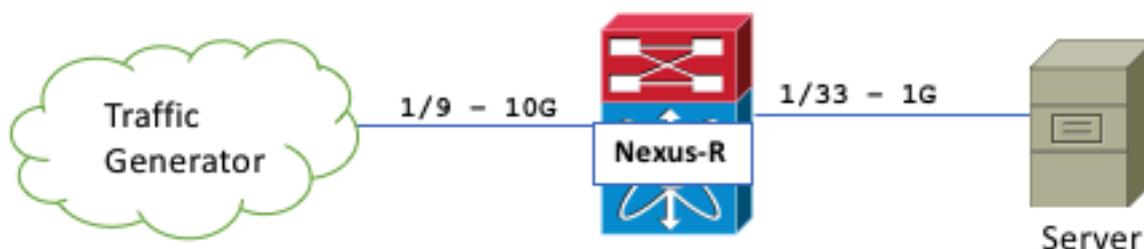
- Datenverkehrsflüsse, die ausgehende Schnittstellen überlasteten (z. B. 10-G-Eingang und 1-G-Ausgang)
- Überbelegter SPAN-Zielport - Gilt für bestimmte Hardwaretypen.

Anwendbare Hardware

PID
N9K-X9636C-R
N9K-X9636Q-R
N9K-X9636C-RX
N9K-X96136YC-R
N3K-C36180YC-R
N3K-C3636C-R

Fehlerbehebung bei Input discard

Häufiges Szenario - Datenverkehrsfluss von 10 G zu 1 G - Konstanter Rückgang:



In diesem Artikel wird der Wert für den Zähler "input discard" und alle internen HW-Leistungsindikatoren, die auf denselben Wert verweisen, geändert, da die Fehler während des


```

CELLS_IN_IPT_CNT          = 606          | CELLS_OUT_IPT_CNT       = 606          |
| ENQ_PKT_CNT              = 1,403,084   |                  |
EGQ_DROP_CNT              = 0            |                  |
| FDT_PKT_CNT              = 1,402,472   |                  |
EGQ_MESHMC_DROP_CNT      = 0            |                  |
| CRC_ERROR_CNT            = 0            |                  |
EGQ_TDM_OVF_DROP_CNT     = 0            |                  |
| CFG_EVENT_CNT            = 606 *       |                  |
|                           = 48,408     |                  |
+-----+-----+-----+-----+-----+
|                           FDT          |                  |
FDR                        |                  |
| IPT_DESC_CELL_COUNTER    = 5,609,892   |                  |
P1_CELL_IN_CNT             = 0            |                  |
| IRE_DESC_CELL_COUNTER    = 0            |                  |
P2_CELL_IN_CNT             = 0            |                  |
|                           = 0            |                  |
P3_CELL_IN_CNT             = 0            |                  |
| TRANSMITTED_DATA_CELLS_COUNTER = 5,609,892 |                  |
CELL_IN_CNT_TOTAL         = 0            |                  |
+-----+-----+-----+-----+-----+
|                           |                  |
/|\                         |                  |
F A C E                       |                  |
|                           |                  |
\|/                         |                  |

```

Wenn `QUEUE_DELETED_PACKET_COUNTER` größer als Null ist, wird angegeben, dass Pakete vom IQM (Ingress Queueing Manager)-Nachwarteschlangen GELÖSCHT wurden. Dies liegt daran, dass eine aktive Warteschlange keine Gutschriften erhält, die auf eine Fehlkonfiguration des Scheduling-Schemas hindeuten würden. Sie können dies über `bcm-shell mod X "getReg IQM_QUEUE_DELETED_PACKET_COUNTER"` überprüfen.

`ENQ_DISCARDED_PACKET_COUNTER` bedeutet, dass Pakete vor der Enqueue verworfen wurden. Sie können diesen Zähleratz auch im BCM sehen (Befehl wird beim Lesen gelöscht):

```

Nexus-R# bcm-shell mod 1 "g iqm_reject_status_bmp" | i i PG|IQM0|IQM1
IQM_REJECT_STATUS_BMP.IQM0[0x1a7]=0x20000000: <VSQF_WRED_STATUS=0,
QNUM_OVF_STATUS=0,PORT_AND_PG_STATUS=1,OCCUPIED_BD_STATUS=0,
IQM_REJECT_STATUS_BMP.IQM1[0x1a7]=0: <VSQF_WRED_STATUS=0,VSQF_MX_SZ_STATUS=0,
PORT_AND_PG_STATUS=0,OCCUPIED_BD_STATUS=0,MULTICAST_ERROR_STATUS=0,

```

Sie können diese immer schnell bemerken, wenn Sie das hardwareinterne Fehlermodul X verwenden (Befehl löscht beim Lesen):

```
Nexus-R# show hardware internal errors module 1
```

```
slot 1
```

```
=====
```

```
|-----|  
| Device:Forwarding ASIC Role:MAC Mod: 1 |  
| Device Statistics Category :: ERROR |  
|-----|  
Instance:0
```

```
IQM
```

```
ENQ_DISCARDED_PACKET_COUNTER = 8,233,862
```

```
Rejects: PORT_AND_PG_STATUS
```

```
Instance:1
```

Schritt 3: Finden Sie heraus, zu welchen ASICs und zu welchen Jericho Port Ihre Input discardFront-Panel-Ports gehören:

```
Nexus-R# show interface hardware-mappings | i i Eth1/9|--|Name|Eth1/33
```

```
HName - Hardware port name. None means N/A
```

```
-----  
Name      Ifindex  Smod Unit HPort HName FPort NPort VPort SrcId  
-----  
Eth1/9    1a001000 0    0    9    xe9   255  8    -1    0    << ASIC 0, Jericho Port 9  
Eth1/33   1a004000 2    1    9    xe9   32   -1    0     << ASIC 1, Jericho Port 9
```

In diesem Beispiel wird Eth1/33 angezeigt. In einem Netzwerk ist der überlastete Ausgangsport noch nicht bekannt.

Schritt 4: Erfahren Sie, welche VOQ- und VOQ-Anschlüsse Ihr Eingangsport hat.

```
Nexus-R# attach module 1
```

```
module-1# show hardware internal jer-usd info voq asic 0 port 9
```

```
+-----+  
|Unit|JerPort| Voq| VoqConn| SE   |   HR   |CreditBal|  
+-----+  
| 0  |    9  | 104|    176| 82213 |    72   | 16a .  |  
+-----+
```

Dieser Befehl zeigt Details für den Flow für das Eingangs-VoQ für einen bestimmten Port an. Außerdem wird hier das aktuelle Guthaben des VoQ angezeigt.

Der VOQ des Ports wird folgendermaßen abgeleitet:

LCs basieren auf 0 - Modul 1 ist 0, Modul 2 ist 1 usw.

Pro LC gibt es 256 System-Port-IDs.

$ID = (LC * \text{System Port ID}) + \text{FP-Nummer}$

$\text{Eth1/9} = (0 * 256) + 9 = 9$

$\text{VOQ-ID} = 32 + (\text{System-Port-ID} * 8)$

$\text{Eth1/9} = 32 + (9 * 8) = 104$

Unsere VOQ für Eth1/9 wird daher 104 betragen, was mit der zuvor erfassten Ausgabe übereinstimmt.

```
module-1# show hardware internal jer-usd ingress-vsq buffer-occupancy front-port 9
+-----+
|                               VSQF BUFFER OCCUPANCY                               |
+-----+
|                               Front port 9                                       |
|max global shared              | 157286 |
|max ocb buffer occupancy       | 0 |
+-----+
|                               COSQ 0                                             |
+-----+
|rate class                     | 4 |
|granted buffers per port       | 3280 |
|shared buffers occupied        | 127792 | <<<<
|granted buffers occupied       | 3280 |
|shared buffer max occupancy    | 127792 | <<<<
+-----+
```

Schritt 5: Überprüfen Sie aus BCM-Sicht, welche Warteschlange speziell nicht leer ist. d. h. überlastet.

```
Nexus-R# bcm-shell mod 1 "diag cosq non_empty_queue"

Core 0:
Ingress VOQs Sizes (format: [queue_id(queue_size)]):
[303(191338496B)] << the Queue ID belongs to your Egress CONGESTED port!

Core 1:
<empty>
```

Schritt 6: Suchen Sie den kongetierten Ausgangsport aus dem Wert für die nicht leere Warteschlange:

Wenn die Warteschlange 303 ist, erinnern Sie sich, dass diese Warteschlangen tatsächlich ein Bereich sind, sodass sie 303 + 7 oder 303-7 sein kann. Es stellt sich die Frage, welcher Port über eine VOQ verfügt, die im Bereich von 296-303 oder alternativ 303-310 übereinstimmt?

Es ist bekannt, dass Warteschlange 7 auf Eth1/9 überlastet ist, also ist 303 tatsächlich der höchste seiner Reichweite, sodass der Bereich von 296-303 eine gut ausgebildete Schätzung ist.

```

module-1# show hardware internal jer-usd info voq asic 1
+-----+
|Unit|JerPort| Voq| VoqConn| SE| HR|CreditBal|
+-----+
| 1| 1| 232| 56| 81957| 8| 3ffff|
| 1| 2| 240| 72| 81989| 16| 3ffff|
| 1| 3| 248| 88| 82021| 24| 3ffff|
| 1| 4| 256| 104| 82053| 32| 3ffff|
| 1| 5| 264| 120| 82085| 40| 3ffff|
| 1| 6| 272| 136| 82117| 48| 3ffff|
| 1| 7| 280| 152| 82149| 56| 3ffff|
| 1| 8| 288| 168| 82181| 64| 3ffff|
| 1| 9| 296| 184| 82213| 72| 3a5| <<< 296 +7 would give us 303
| 1| 10| 304| 200| 82245| 80| 3ffff| << It cannot be this one as 303 is not included
| 1| 11| 312| 216| 82277| 88| 3ffff|
<snip>

```

Gleiches für Basis 0 anzeigen - Hier nicht für Kurzfristigkeit dargestellt; in der Voq-Spalte wird angezeigt, dass Ihr Interesse nicht an diesem ASIC liegt.

Beachten Sie einige Punkte in der obigen Ausgabe:

- Unser ausgehender überlasteter Port befindet sich auf ASIC 1.
- Unser ausgehender überlasteter Port hat einen VOQ von 296, und 303 entspricht Warteschlange 7 an diesem Port.
- Beachten Sie die Spalte "Credit Balance" (Kontostand) - Es sind nur sehr wenige Gutschriften auf dieser Schnittstelle verfügbar, die gewährt werden müssen. Deshalb beginnt unsere Eingangs-Eth1/9-Pufferung.

Schritt 7: Überprüfen Sie, welcher Frontblenden-Port sich im ASIC 1 befindet, und ordnen Sie ihn Jericho Port 9 entsprechend Ihrer vorherigen Suche zu.

```

Nexus-R# show interface hardware-mappings | i i Eth1/9|--|Name|Eth1/33
      HName - Hardware port name. None means N/A
-----
Name      Ifindex  Smod Unit HPort HName FPort NPort VPort SrcId
-----
Eth1/9    1a001000 0    0    9    xe9   255  8    -1    0    << ASIC 0, Jericho Port 9
Eth1/33   1a004000 2    1    9    xe9   32   -1    0     << ASIC 1, Jericho Port 9

```

An diesem Punkt haben Sie den überlasteten Ausgangsport gefunden. Stellen Sie fest, ob ein falsches Hochladen im Netzwerk vorliegt, Sie haben das SPAN konfiguriert, und Ihr Zielpport ist 1G, wenn Sie eine oder mehrere 10G-Schnittstellen beziehen, oder ob es sich um einen Engpass/ein Designproblem handelt.

Zusätzliche Befehle

Diese sind weiter fortgeschritten - werden unter normalen Szenarien nicht benötigt, um den über Egress Congested verbundenen Port zu finden.

```

attach module X
show hardware internal jer-usd tm_debug asic <slot> module <module>
show hardware internal jer-usd info voq [ asic <instance> ] [ port <port> ] [ ]
show hardware internal jer-usd info non-empty voq asic [ <instance> ] [ ]

```

```

show hardware internal jer-usd info voq-profile { QueueThreshold drop_p <dp> | OCBThreshold } [
asic <instance> ] [ port<port> ] [ ]
show hardware internal jer-usd info voq-connector front-port <port> [ ]
show hardware internal jer-usd stats vsq { front-port <port> | inband asic <slot> | recycle-port
<port> asic <slot> }
show hardware internal jer-usd ingress-vsq buffer-occupancy front-port <port>
show hardware internal jer-usd info IQM { counter | rate } asic <instance> dst-port <port> [
interval <int> ] [ ]
show hardware internal jer-usd info SCH { counter | rate } asic <instance> dst-port <port> [
interval <int> ] [ ]

```

```

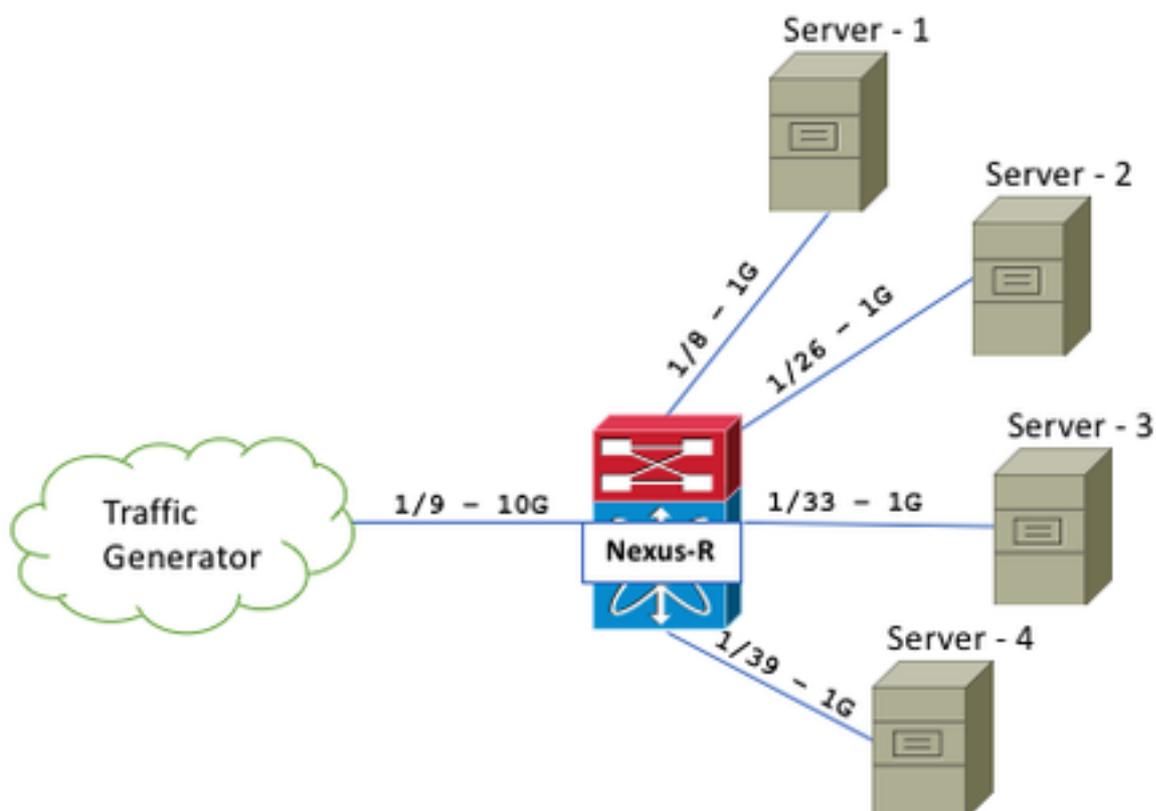
bcm-shell mod X
diag cosq print_flow_and_up dest_id=<flow_id>
diag cosq voq id=<voqid> detailed=1
diag cosq qpair e2e ps=<id>
cosq conn ing
cosq conn egr
dump IPS_CR_BAL_TABLE <voqID>
getReg IQM_QUEUE_MAXIMUM_OCCUPANCY_QUEUE_SIZE

```

Zusätzliche Labortests:

Schritt 1: Input discard mit mehreren konvergierten Ausgangsschnittstellen.

In dieser Topologie sendet der Traffic Generator 2 G Datenverkehr an jeden Server:



Überprüfen Sie schnell, welche Warteschlangen nicht leer sind. Beachten Sie, dass es 4 gibt:

```
Nexus-R# bcm-shell mod 1 "diag cosq non_empty_queue"
```

```
Core 0:
```

```
Ingress VOQs Sizes (format: [queue_id(queue_size)]):
```

```
[103(29475840B)] [247(29379584B)] [303(56452096B)] [351(76020736B)]
```

Bestimmen Sie, zu welchen Schnittstellen diese Warteschlangen gehören - Aktivieren Sie zuerst ASIC 0 (es wird nur mit einer Schnittstelle gezeigt):

```
module-1# show hardware internal jer-usd info voq asic 0
```

```
+-----+
|Unit|JerPort| Voq| VoqConn| SE| HR|CreditBal|
+-----+
| 0| 1| 40| 48| 81957| 8| 3ffff|
| 0| 2| 48| 64| 81989| 16| 3ffff|
| 0| 3| 56| 80| 82021| 24| 3ffff|
...
| 0| 8| 96| 160| 82181| 64| 7b| << 96 + 7 = 103, this is port Eth1/8
<snip>
```

```
`show interface hardware-mappings`
```

```
-----
Name      Ifindex  Smod Unit HPort HName NPort VPort SrcId
-----
Eth1/8    1a000e00 0 0 8 xe8 7 -1 0
```

Wiederholen Sie den gleichen Prozess für die anderen drei Warteschlangenwerte: 247, 303 und 351.

Schritt 2: Input discard aufgrund von SPAN.

Einstellen von Eth1/33 als SPAN-Zielport und Festlegen von Eth1/9 als SPAN-Quellport in RX-Richtung

```
Nexus-R# show run mon
```

```
monitor session 1
description SPAN TEST INPUT DISCARDS
source interface Ethernet1/9 rx
destination interface Ethernet1/33
no shut
```

```
Nexus-R# show int e1/9 | i i input.disc
0 input with dribble 9314306 input discard
```

Schritt 3: Input discard aufgrund von Traffic Hair Pin.

Senden von Paketen mit SRC 10.10.10.10 und DEST 192.168.10.10, wobei Eth1/9 in 10.10.10.1/24 ist - dies führt nicht zu einer Input discard. Dieser Zähler wird jedoch angezeigt:

```
Nexus-R# bcm-shell mod 1 "diag counters g"
```

```
| / \
```

```
| J E R I C H O N E T W O R K I N T E R F A C E |
\|/ |
```

```
-----+-----
-----+-----
_PACKET_COUNTER = 0 | DELETED_PKT_CNT = 12,027,201 |
| | Discards: INVALID_OTM SRC_EQUAL_DEST
```

Schritt 4: Senden von Paketen mit einer Ziel-IP, die unbekannt ist.

Senden Sie Pakete mit SRC 10.10.10.10 und DEST 192.168.10.10, wobei Eth1/9 im Subnetz 10.10.10.1/24 und Eth1/33 im Subnetz 172.16.0.1/30 liegt. - Keine Drop-Zähler, keine input discard, auch wenn das Ziel unbekannt ist.

Schritt 5: Input discard, während ein Zugangs-/Trunk-Port in den STP-Weiterleitungsstatus wechselt

Senden von Paketen, bei denen Eth1/9 nur ein Wide-Trunk (oder Access-Port) ist - Dieser wird als Input discard registriert, während der Port in einen STP-Weiterleitungsstatus wechselt.

```
Nexus-R(config)# int e1/9
Nexus-R(config-if)# switchport mode trunk
Nexus-R# bcm-shell mod 1 "diag counters g" | i i --|IQM|ENQ_DISCARD|Rejects
```

```
-----+-----
-----+-----
-----+-----
-----+-----
-----+-----
-----+-----
PQP_MC_PKT_CNT = 1,678,949 |
| IQM |
PQP_DSCRD_MC_PKT_CNT = 11,369,033 |
| ENQ_DISCARDED_PACKET_COUNTER = 1,289,182 |
DELETED_PKT_CNT = 11,369,081 |
| Rejects: QUEUE_NOT_VALID_STATUS |
Discards: SRC_EQUAL_DEST |
```

```
Nexus-R# show span int e1/9
```

```
Vlan Role Sts Cost Prio.Nbr Type
-----
VLAN0001 Desg BLK 2 128.9 P2p
VLAN0010 Desg BLK 2 128.9 P2p
<snip>
```

QUEUE_NOT_VALID_STATUS ist ein Tropfen aufgrund der Entscheidung des Packet Processor (PP), einen Eintrag zu verwerfen, oder ein ungültiges Ziel, das von den Packet Processor (PP) Blocks empfangen wurde.

Schritt 6: Input discard aufgrund von Eth1/9, die die Leitungsgeschwindigkeit überschreiten.

Sendet man 10G+ an Eth1/9, würde dies zu einem anderen Tropftyp führen, da man Eth1/9 zuerst ausfüllt - Zählt immer noch als Input discard:

bcm-shell.0> **diag counters g**

```

/|\
|
R F A C E |
|
|
+-----+
|
|
| RX_TOTAL_BYTE_COUNTER = 53,913,106,009 |
TX_TOTAL_BYTE_COUNTER = 1,164,231 |
| RX_TOTAL_PKT_COUNTER = 54,145,395 |
TX_TOTAL_PKT_COUNTER = 17,029 |
| RX_TOTAL_DROPPED_EOPS = 0 |
|
+-----+
|
| IRE
EPNI |
| CPU_PACKET_COUNTER = 17,010 |
|
| NIF_PACKET_COUNTER = 54,145,476 |
EPE_BYTES_COUNTER = 5,721,307 |
| OAMP_PACKET_COUNTER = 0 |
EPE_PKT_COUNTER = 50,703 |
| OLP_PACKET_COUNTER = 0 |
EPE_DSCRD_PKT_CNT = 0 |
| RCY_PACKET_COUNTER = 16,837 |
|
| IRE_FDT_INTRFACE_CNT = 0 |
|
+-----+
|
| IDR
EGQ |
|
| MMU_IDR_PACKET_COUNTER = 54,128,577 |
FQP_PACKET_COUNTER = 50,703 |
| IDR_OCB_INTERFACE_COUNTER = 0 |
PQP_UNICAST_PKT_CNT = 50,683 |
|
PQP_DSCRD_UC_PKT_CNT = 0 |
|
PQP_UC_BYTES_CNT = 5,216,716 |
+-----+
PQP_MC_PKT_CNT = 20 |
| IQM
PQP_DSCRD_MC_PKT_CNT = 20 |
|
PQP_MC_BYTES_CNT = 2,079 |
| ENQUEUE_PKT_CNT = 5,463,323 |
EHP_UNICAST_PKT_CNT = 50,683 |
| DEQUEUE_PKT_CNT = 5,594,400 |
EHP_MC_HIGH_PKT_CNT = 20 |
| DELETED_PKT_CNT = 0 |
EHP_MC_LOW_PKT_CNT = 0 |
| ENQ_DISCARDED_PACKET_COUNTER = 48,716,055 |
DELETED_PKT_CNT = 40 |

```

| **Rejects: VOQ_MX_QSZ_STATUS** |

|
<snip>