

# Identificar e Solucionar Problemas de Quedas de Entrada no IOS XR

## Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Informações de Apoio](#)

[Problema: Incremento na queda de entrada](#)

[Quedas do controlador](#)

[Endereço de Controle de Acesso Médio \(DMAC - Medium Access Control Address\) de destino desconhecido ou VLAN dot1q](#)

[Pacotes ignorados devido a protocolo de nível superior não reconhecido](#)

[Quedas NP no ASR 9000](#)

[Netio](#)

## Introduction

Este documento descreve como solucionar problemas de quedas de entrada na interface em roteadores XR.

## Prerequisites

### Requirements

Não existem requisitos específicos para este documento.

### Componentes Utilizados

Este artigo aborda os roteadores da série ASR 9000, os roteadores da série CRS e os roteadores da série GSR 12000.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. Se a rede estiver ativa, certifique-se de que você entenda o impacto potencial de qualquer comando.

## Informações de Apoio

Os descartes de entrada no IOS XR têm um significado completamente diferente dos descartes de entrada no IOS. Ele pode confundir você quando migra o IOS para o IOS XR e começa a ver seus contadores de queda de entrada no show interface.

No IOS, uma queda de entrada foi causada pela fila de entrada da interface que fica cheia. Isso significa que muitos pacotes foram lançados para a CPU para switching de processo e ela não foi capaz de tratá-los com rapidez suficiente. A fila de entrada é construída até ficar cheia e haver alguns descartes.

No IOS XR, não há uma definição estrita de um drop de entrada. Então, basicamente, cabe aos desenvolvedores de um componente decidir se querem incrementar o contador de queda de entrada quando decidem descartar um pacote. O principal aqui é que em algum ponto do código, o roteador decide descartar o pacote, o que significa que é provável que o roteador não deveria encaminhar esse pacote e o roteador decide conscientemente descartá-lo. Portanto, isso não está relacionado ao congestionamento, como no IOS. No entanto, trata-se de um pacote que foi recebido pelo roteador e que não deveria ser encaminhado, portanto o roteador decidiu descartá-lo e, muito provavelmente, não é motivo para alarme. Embora, você não possa dizer se é algo com que se preocupar ou não até que você tenha entendido completamente o tipo de pacotes que estão incrementando o contador de queda de entrada e isso não é tão simples, infelizmente.

Examples:

- Um roteador XR é conectado a um switch que envia alguns pacotes BPDUs (bridge protocol data unit) e UDLD. O roteador XR não tem spanning-tree nem UDLD configurados em suas interfaces de camada 3, portanto ele apenas descarta esses quadros e incrementa o contador de descartes de entrada em show interface. Nesse caso, não há nada com que se preocupar, pois é a coisa certa a fazer para descartar esses quadros, pois os recursos não estão configurados.
- Um ASR 9000 tem uma entrada do Cisco Express Forwarding (CEF) programada incorretamente devido a um bug, de modo que não aponta para uma adjacência válida. Nesse caso, o processador de rede da placa de linha (LC) ASR 9000 percebe que o roteador perde um loadinfo e incrementa um contador de queda do processador de rede (NP) que é carregado no contador de queda de entrada da interface.

Quando as quedas de entrada são relatadas, o problema é descobrir se essas são quedas legítimas, como no exemplo 1, ou a consequência de um problema, como no exemplo 2.

## Problema: Incremento na queda de entrada

Este documento enumera os motivos para os descartes de entrada que são incrementados e como verificar se é esse o motivo:

### Quedas do controlador

Runts, Sequência de Verificação de Quadro (FCS), abortos, estouros de First Input First Output (FIFO), quedas gigantes de Pacote sobre SDH/SONET (POS).

```
RP/0/RP0/CPU0:equinox#show controllers poS 0/2/0/0 framer statistics
POS Driver Internal Cooked Stats Values for port 0
=====
Rx Statistics                               Tx Statistics
-----
```

Total Bytes:	71346296	Total Bytes:	67718333
Good Bytes:	71346296	Good Bytes:	67718333
Good Packets:	105385	Good Packets:	67281
Aborts:	0	Aborts:	0
FCS Errors:	0	Min-len errors:	0
Runts:	0	Max-len errors:	0
FIFO Overflows:	0	FIFO Underruns:	0
Giants:	0		
Drops:	0		

RP/0/RP0/CPU0:equinox#

Para uma interface ethernet (gige, tengige...), verifique algo como:

### show controllers gigabitEthernet 0/0/0/18 stats

Veja se há um contador nas estatísticas do controlador que incrementa na mesma taxa que o contador de queda de entrada em show interface. Alguns desses contadores de erro também devem estar em show interface.

### Endereço de Controle de Acesso Médio (DMAC - Medium Access Control Address) de destino desconhecido ou VLAN dot1q

Pacotes com um endereço MAC de destino que não seja o da interface ou com uma rede local virtual (VLAN) não correspondida por uma subinterface. Isso pode acontecer quando há alguma inundação em um domínio L2 de endereços MAC unicast desconhecidos, de modo que o roteador XR conectado a esse domínio L2 recebe quadros com um endereço MAC de destino que não é um de seus controladores. Também é possível quando um roteador IOS está enviando keepalives de ethernet em sua interface gige, portanto, esses keepalives estão incrementando as quedas de entrada no roteador XR, pois não têm o endereço mac de destino do roteador XR. Ou quando a interface está conectada a outro dispositivo que tem mais vlans/subinterfaces dot1q configuradas como no roteador XR, de modo que o roteador XR receba quadros com uma marca dot1q desconhecida.

Em um PLIM (Physical Layer Interface Modules) fixo do CRS, você pode encontrar esses descartes em:

```
RP/0/RP0/CPU0:pixies-uk#sh contr plim ASIC statistics interface tenGigE 0/1/0/3 location 0/1/CPU0
```

Wed Aug 22 16:07:47.854 CEST

Node: 0/1/CPU0

TenGigE0/1/0/3 Drop

```
-----
RxFIFO Drop      : 0
PAR Tail Drop    : 0
PAR Err Drop    : 0
Invalid MAC Drop : 86
TxFIFO Drop    : 0
Invalid VLAN Drop : 11
```

Ou nas estatísticas do controlador de tengige ou gige:

```
RP/0/RP0/CPU0:pixies-uk#sh contr ten 0/1/0/3 stats
Wed Aug 22 16:22:42.059 CEST
Statistics for interface TenGigE0/1/0/3 (cached values):
```

Ingress:

```
Input drop overrun          = 0
Input drop abort            = 0
Input drop invalid VLAN    = 11
Input drop invalid DMAC    = 0
Input drop invalid encap   = 0
Input drop other            = 86
```

**Observação:** o bug [CSCub74803](#) existe, o **Input drop other** é incrementado em vez de **Input drop invalid DMAC** pelo menos no PLIM fixo de tengige de 8 portas do CRS.

Para Adaptadores de Porta Compartilhada (SPAs - Shared Port Adapters) (CRS, XR 12000), os pacotes com MAC inválido seriam descartados pelo SPA l2-tcam para que você possa encontrar esses descartes em **show controllers TenGigE a/b/c/d all**:

```
Input drop other          = 107
```

```
l2-tcam Invalid DA Drops: 107
```

Em um ASR 9000, os contadores **Input drop invalid DMAC** e **Input drop other** nas estatísticas do controlador não são incrementados. Assim, a maneira de reconhecer essas quedas no ASR 9000 é encontrar o NP que manipula a interface com as quedas de entrada:

```
RP/0/RSP0/CPU0:obama#sh int gig 0/0/0/30 | i "input drops"
Wed Aug 22 16:55:52.374 CEST
  1155 packets input, 156256 bytes, 1000 total input drops
RP/0/RSP0/CPU0:obama#sh contr np ports all location 0/0/CPU0
Wed Aug 22 16:56:01.385 CEST
```

```
Node: 0/0/CPU0:
```

```
-----
NP Bridge Fia                      Ports
--
0 0      0 GigabitEthernet0/0/0/30 - GigabitEthernet0/0/0/39
1 0      0 GigabitEthernet0/0/0/20 - GigabitEthernet0/0/0/29
2 1      0 GigabitEthernet0/0/0/10 - GigabitEthernet0/0/0/19
3 1      0 GigabitEthernet0/0/0/0 - GigabitEthernet0/0/0/9
RP/0/RSP0/CPU0:obama#
```

Assim, você pode ver que a interface gig 0/0/0/30 é manipulada pelo NP 0 em 0/0/CPU0. Vamos verificar os contadores NP de NP0 em 0/0/CPU0:

```
RP/0/RSP0/CPU0:obama#sh contr np counters np0 location 0/0/CPU0
```

Wed Aug 22 16:56:19.883 CEST

Node: 0/0/CPU0:

-----  
Show global stats counters for NP0, revision v3

Read 26 non-zero NP counters:

Offset	Counter	FrameValue	Rate (pps)
22	PARSE_ENET_RECEIVE_CNT	1465	0
23	PARSE_FABRIC_RECEIVE_CNT	2793	0
24	PARSE_LOOPBACK_RECEIVE_CNT	2800	0
28	MODIFY_FABRIC_TRANSMIT_CNT	80	0
29	MODIFY_ENET_TRANSMIT_CNT	1792	0
32	RESOLVE_INGRESS_DROP_CNT	1000	0
35	MODIFY_EGRESS_DROP_CNT	1400	0
36	MODIFY_MCAST_FLD_LOOPBACK_CNT	1400	0
38	PARSE_INGRESS_PUNT_CNT	465	0
39	PARSE_EGRESS_PUNT_CNT	155	0
45	MODIFY_RPF_FAIL_DROP_CNT	1400	0
53	PARSE_LC_INJECT_TO_FAB_CNT	80	0
54	PARSE_LC_INJECT_TO_PORT_CNT	864	0
57	PARSE_FAB_INJECT_UNKN_CNT	155	0
67	RESOLVE_INGRESS_L3_PUNT_CNT	465	0
69	RESOLVE_INGRESS_L2_PUNT_CNT	464	0
70	RESOLVE_EGRESS_L3_PUNT_CNT	1400	0
93	CDP	464	0
95	ARP	1	0
109	DIAGS	154	0
221	PUNT_STATISTICS	9142	1
223	PUNT_DIAGS_RSP_ACT	155	0
225	PUNT_DIAGS_RSP_STBY	155	0
227	NETIO_RP_TO_LC_CPU_PUNT	155	0
<b>373</b>	<b>L3_NOT_MYMAC</b>	<b>1000</b>	<b>0</b>
565	INJECT_EGR_PARSE_PRRT_PIT	928	0

RP/0/RSP0/CPU0: obama#

Portanto, L3\_NOT\_MYMAC no contador NP significa que o roteador recebeu um quadro em uma interface de Camada 3 com um endereço MAC de destino que não é uma das interfaces. E o roteador o descarta como esperado e isso é relatado como descartes de entrada em show interface.

No ASR 9000 para pacotes recebidos com uma VLAN dot1q não configurada em uma subinterface do ASR 9000, o contador **Input drop unknown 802.1Q** não é incrementado em **show controllers gigabitEthernet 0/0/0/30 stats**. O procedimento é o mesmo que acima para o DMAC desconhecido: identificar qual NP manipula as interfaces e, em seguida, verificar esses contadores NP. Você vê que o contador NP **UIDB\_TCAM\_MISS\_AGG\_DROP** está sendo incrementado nesse caso.

## Pacotes Ignorados devido a Protocolo de Nível Superior Não Reconhecido

Esse é simples de detectar, pois há um contador para essas quedas, uma linha abaixo das quedas de entrada em show interface:

```
RP/0/RSP0/CPU0:obama#sh int gig 0/0/0/18
Wed Aug 22 17:14:35.232 CEST
GigabitEthernet0/0/0/18 is up, line protocol is up

 5 minute input rate 4000 bits/sec, 0 packets/sec
 5 minute output rate 5000 bits/sec, 0 packets/sec
 7375 packets input, 6565506 bytes, 1481 total input drops
 1481 drops for unrecognized upper-level protocol
```

Você pode ver aqui que todas as quedas de entrada foram causadas por um protocolo de nível superior não reconhecido.

Isso significa que os pacotes foram recebidos com um protocolo Ethernet no qual o roteador não está interessado. Isso significa que um recurso está configurado no vizinho (ou em um host conectado ao domínio da camada 2 conectado a essa interface) para que ele envie quadros com um protocolo não configurado no roteador XR.

Exemplos: BPDUs, Intermediate System to Intermediate System (ISIS), Connection less Network Protocol (CLNP), IPv6, UDLD, Cisco Discovery Protocol (CDP), VLAN Trunking Protocol (VTP), Dynamic Trunking Protocol (DTP), Link Layer Discovery Protocol (LLDP) etc....

Quando esses recursos não são configurados na interface XR, a caixa XR os descarta conforme esperado. Para descobrir que tipos de quadros estão incrementando esse contador, você terá que comparar quais recursos estão habilitados no roteador XR com os recursos habilitados no vizinho (pode ser um roteador ou um switch), ou os recursos habilitados em todos os dispositivos conectados aos domínios de camada 2 conectados a essa interface (muito menos fácil). Se o roteador XR estiver conectado a um switch, você poderá tentar um span nesse switch dos pacotes que ele envia ao roteador XR na interface com quedas de entrada. Consulte:

[ASR9000/XR: descartes para erro de protocolo de nível superior não reconhecido](#)

## Quedas NP no ASR 9000

Os contadores de descarte no Network Process (NP) do ASR 9000 são relatados como descartes de entrada quando se aplicam a um pacote recebido em uma interface e descartado. Isso não acontece para quedas do Packet Switch Engine (PSE) no CRS e no 12000 XR: elas não são contadas como quedas de entrada.

Portanto, se você tiver quedas de entrada em um ASR 9000 e elas não corresponderem a um desses motivos, então você faria um **show controllers np ports all location 0/<x>/CPU0** para encontrar o NP que manipula a interface com quedas de entrada e, em seguida, verifique seus contadores NP com **show contr np counters np<y> location 0/<x>/CPU0**.

Você pode canalizar a saída apenas para manter contadores DROP com um comando como **sh contr np counters np<y> location 0/<x>/CPU0 | i DROP** mas isso pode ser perigoso, pois às vezes um contador de queda não tem DROP em seu nome. Você viu um bom exemplo com L3\_NOT\_MYMAC. Então talvez pipe para **DROP|DISCARD|NOT|EXCD**.

Você pode limpar os contadores de interface e os contadores NP com **clear controller np counters np<y> location 0/<x>/CPU0** aproximadamente ao mesmo tempo para descobrir qual contador NP incrementa na mesma taxa que a entrada cai.

Por exemplo, você obtém IPV4\_PLU\_DROP\_PKT nos contadores NP, o que significa que a entrada CEF/PLU está dizendo que o pacote precisa ser descartado. Você não tem uma rota padrão e os inalcançáveis estão desabilitados para que os pacotes não correspondam a uma rota mais específica ao atingirem o manipulador CEF padrão, que é uma entrada de queda.

Se você encontrar um contador de queda no NP que possa explicar as quedas de entrada à medida que elas incrementam na mesma taxa, mas o contador de queda NP não é muito autoexplicativo, você pode navegar nesta página para tentar entender o que o contador significa: [ASR9000/XR: Solucionar problemas de quedas de pacotes e compreender contadores de quedas NP](#)

**Observação:** a página de Xander em fóruns de suporte contém as razões da queda para a primeira geração de placas de linha (Trident) e há novos nomes de contador para as placas de linha de nova geração (Typhoon)... mas com base no nome, você deve ser capaz de encontrar um nome de contador semelhante ao Trident.

## Netio

Você pode coletar um **show netio idb <int>** e isso fornece a você o drop de entrada da interface e os contadores de drop do nó netio:

```
RP/0/RP0/CPU0:ipc-lsp690-r-ca-01#show netio idb gigabitEthernet 0/2/0/1
```

```
GigabitEthernet0/2/0/1 (handle: 0x01280040, nodeid:0x21) netio idb:
```

```
-----
name:                               GigabitEthernet0_2_0_1
interface handle:                    0x01280040
interface global index:              3
physical media type:                 30
dchain ptr:                          <0x482e0700>
echain ptr:                          <0x482e1024>
fchain ptr:                          <0x482e13ec>
driver cookie:                       <0x4829fc6c>
driver func:                          <0x4829f040>
number of subinterfaces:             4096
subblock array size:                 7
DSNCF:                               0x00000000
interface stats info:
  IN  unknown proto pkts:            0
  IN  unknown proto bytes:           0
  IN  multicast pkts:                 0
  OUT multicast pkts:                 0
  IN  broadcast pkts:                 0
  OUT broadcast pkts:                 0
  IN  drop pkts:                   0 <===== cleared when added to input drop counter !!!>
  OUT drop pkts:                      0
  IN  errors pkts:                    0
  OUT errors pkts:                     0
```

### Chains

```
-----
Base decap chain:
  ether                               <30> <0xfd018cd8, 0x482c736c> < 0, 0>
```

### Protocol chains:

```
-----
<Protocol number> (name) Stats
Type Chain_node      <caps num> <function, context> <drop pkts, drop bytes>
<snip>
<13> (mpls) Stats IN: 204 pkts, 23256 bytes; OUT: 0 pkts, 0 bytes
  Encap:
```

```

mpls                <25> <0xfcc7ddbc, 0x00000000> <      0,      0>
ether               <30> <0xfd0189b4, 0x482c736c> <      0,      0>
l2_adj_rewrite     <86> <0xfcaa997c, 0x4831a2e8> <      0,      0>
pcn_output         <54> <0xfd0561f0, 0x48319f04> <      0,      0>
q_fq               <43> <0xfd05f4b8, 0x48320fec> <      0,      0>
txm_nopull        <60> <0xfcadba38, 0x4824c0fc> <      0,      0>
Decap:
pcn_input          <55> <0xfd0561f0, 0x4830ba8c> <      0,      0>
q_fq_input        <96> <0xfd05f330, 0x48312c7c> <      0,      0>
  mpls            <25> <0xfcc7b2b8, 0x00000000> <    152,    17328>
Fixup:
l2_adj_rewrite     <86> <0xfcaa945c, 0x00000000> <      0,      0>
pcn_output         <54> <0xfd0561f0, 0x48319f04> <      0,      0>
q_fq               <43> <0xfd05f4b8, 0x48320fec> <      0,      0>
txm_nopull        <60> <0xfcadba38, 0x4824c0fc> <      0,      0>

```

As quedas no nó Multi-Protocol Label Switching (MPLS) aqui podem ocorrer devido ao tempo de vida (TTL) de MPLS expirado (no caso de um loop ou quando o cliente faz um traceroute) ou à fragmentação necessária e ao conjunto de bits Do Not Fragment (DF) por exemplo. Você pode executar **debug mpls packet drop** e **debug mpls error** com o local da interface para tentar descobrir que tipo de pacote está incrementando esse contador.

Pacotes multicast pontuados. Quando você vê netio **IN drop pkts**, mas nenhum nó netio abaixo com algumas quedas que poderiam explicar os **IN drop pkts**, então isso pode ser mcast punted packets e você pode habilitar **deb mfib netio drop** para descobrir que tipo de pacotes



Sobre esta tradução

A Cisco traduziu este documento com a ajuda de tecnologias de tradução automática e humana para oferecer conteúdo de suporte aos seus usuários no seu próprio idioma, independentemente da localização.

Observe que mesmo a melhor tradução automática não será tão precisa quanto as realizadas por um tradutor profissional.

A Cisco Systems, Inc. não se responsabiliza pela precisão destas traduções e recomenda que o documento original em inglês ([link fornecido](#)) seja sempre consultado.