

Verstehen von Pufferfehlern und -fehlern

Inhalt

[Einführung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[Konventionen](#)

[Pufferfehler und -fehler](#)

[Puffer-Pools](#)

[Konfigurationsbefehl für Puffer](#)

[Zusätzliche Show-Befehle](#)

[Zugehörige Informationen](#)

Einführung

In diesem Dokument werden Pufferüberläufe und Fehlfunktionen auf dem Routingprozessor (RP) behandelt.

Voraussetzungen

Anforderungen

Für dieses Dokument bestehen keine speziellen Anforderungen.

Verwendete Komponenten

Dieses Dokument ist nicht auf bestimmte Software- und Hardwareversionen beschränkt.

Die Informationen in diesem Dokument wurden von den Geräten in einer bestimmten Laborumgebung erstellt. Alle in diesem Dokument verwendeten Geräte haben mit einer leeren (Standard-)Konfiguration begonnen. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die potenziellen Auswirkungen eines Befehls verstehen.

Konventionen

Weitere Informationen zu Dokumentkonventionen finden Sie unter [Cisco Technical Tips Conventions](#) (Technische Tipps zu Konventionen von Cisco).

Pufferfehler und -fehler

Der RP teilt seinen Prozessorspeicher in Pools auf. Jeder Pool enthält mehrere Speicherblöcke gleicher Größe. Diese Speicherblöcke werden als Puffer bezeichnet.

Puffer-Pools

Es gibt sechs Puffer-Pools:

- Klein - 104 Byte Puffer
- Mittel - 600-Byte-Puffer
- Big - 1524-Byte-Puffer
- VeryBig - 4520 Byte Puffer
- Large - 5024 Byte Puffer
- Huge - Puffer mit 18.024 Byte

Wenn beispielsweise ein Schnittstellenprozessor ein 20-Byte-Paket an den RP übergeben muss, "fragt" er nach einem kleinen Puffer. Wenn ein Schnittstellenprozessor ein 500-Byte-Paket an den RP übergeben muss, fordert er einen mittleren Puffer usw.

Hinweis: Der Schnittstellenprozessor muss einen Puffer einer bestimmten Größe anfordern.

Wenn der Schnittstellenprozessor nach einem Puffer fragt, geschieht Folgendes:

- Wenn innerhalb des angeforderten Pools ein freier Puffer vorhanden ist, wird der Puffer gewährt. Andernfalls generiert die Anforderung einen "verpassten" und der Pufferalgorithmus versucht, mehr Puffer für diesen Pool zu "erstellen".
- Wenn IOS keinen kleinen Puffer abrufen kann, wird das Paket nicht verworfen. Er erhöht den ausgefallenen Zähler und gelangt zum Puffer der nächsten Ebene, dem Mittelpuffer, und fordert dort einen Puffer an. Wenn es keinen Mittelpuffer abrufen kann, wird der Puffer der nächsten Ebene angefordert, der ein großer Puffer ist. Dieser Prozess wird fortgesetzt, bis er den riesigen Puffer-Pool erreicht. Wenn er keinen großen Puffer erhält, verwirft er das Paket.
- Wenn Sie das IBM-Feature-Set verwenden, verursacht ein Ausfall fast immer einen Ausfall.
- Obwohl die IBM-Funktionen prozessgesteuert sein können, wird der Code zum Abrufen eines Puffers für die Übergabe eines Pakets von einer Schnittstelle an den RP auf Unterbrechungsebene ausgeführt.
- Puffer können nicht auf Unterbrechungsebene erstellt werden. Daher stellt ein Versäumnis seine Anforderung nach mehr Puffern an den RP in die Warteschlange.
- Da vor Ort kein zusätzlicher Puffer erstellt werden kann, schlägt die Pufferanforderung fehl und das Paket wird verworfen.

Pufferfehler sind einer der häufigsten Gründe für Paketverluste. Wenn Paketverluste aufgrund eines Pufferausfalls auftreten, geschieht Folgendes:

- Nach einem Pufferausfall hat der RP eine ausstehende Anforderung, mehr Puffer der entsprechenden Größe für den jeweiligen Pool zu erstellen.
- Während der RP die Anforderung der erstellten Puffer bearbeitet, können weitere Fehler im Pool auftreten.
- Der RP kann sogar verhindern, dass mehr Puffer erstellt werden, da Speicherbeschränkungen im System vorliegen, wenn die zusätzlichen Puffer erforderlich sind.
- Im Wesentlichen könnte der Vorgang zum Erstellen von Puffern mehrere Mikrosekunden dauern, in denen Pakete aufgrund des Pufferknappes fortwährend verworfen werden.
- Wenn Puffer so schnell verwendet werden, wie sie erstellt werden, könnte der RP gezwungen

sein, mehr Zeit für die Puffer-Erstellung als für die Paketverarbeitung zu verbringen.

- Dies kann dazu führen, dass der RP beginnt, Pakete so schnell zu verwerfen, dass die Leistung abnimmt und Sitzungen verloren gehen.

Glücklicherweise sind Pufferfehler, wie in diesem Dokument beschrieben, nicht schwer zu identifizieren und zu beheben. Diese **Ausgabe des Befehls "Puffer"** zeigt den aktuellen Zustand der Pufferpools des Routers an:

```
dspu-7k#show buffers
```

```
Buffer elements:
```

```
500 in free list (500 max allowed)
2370 hits, 0 misses, 0 created
```

```
Public buffer pools:
```

```
Small buffers, 104 bytes (total 16, permanent 10):
```

```
11 in free list (0 min, 10 max allowed)
1770 hits, 33 misses, 22 trims, 28 created
9 failures (0 no memory)
```

```
Middle buffers, 600 bytes (total 90, permanent 90):
```

```
89 in free list (10 min, 200 max allowed)
590 hits, 0 misses, 0 trims, 0 created
0 failures (0 no memory)
```

```
Big buffers, 1524 bytes (total 90, permanent 90):
```

```
90 in free list (5 min, 300 max allowed)
126 hits, 0 misses, 0 trims, 0 created
0 failures (0 no memory)
```

```
VeryBig buffers, 4520 bytes (total 10, permanent 10):
```

```
10 in free list (0 min, 300 max allowed)
50 hits, 0 misses, 0 trims, 0 created
0 failures (0 no memory)
```

```
Large buffers, 5024 bytes (total 10, permanent 10):
```

```
10 in free list (0 min, 30 max allowed)
0 hits, 0 misses, 0 trims, 0 created
0 failures (0 no memory)
```

```
Huge buffers, 18024 bytes (total 2, permanent 0):
```

```
0 in free list (0 min, 13 max allowed)
2 hits, 2 misses, 0 trims, 2 created
0 failures (0 no memory)
```

In der Ausgabe **show buffers**:

- Die **Gesamtzahl** der Puffer im Pool, einschließlich verwendeter und nicht verwendeter Puffer, wird angegeben.
- **Permanent** gibt die permanente Anzahl der zugewiesenen Puffer im Pool an. Diese Puffer befinden sich immer im Pool und können nicht abgeschnitten werden.
- In der **freien Liste** wird die Anzahl der Puffer angegeben, die derzeit im Pool verfügbar sind.
- **Min** gibt die Mindestanzahl an Puffern an, die der RP in der freien Liste beibehalten sollte: Der **min**-Parameter dient dazu, die Nachfrage nach Puffern aus dem Pool jederzeit vorherzusehen. Wenn die Anzahl der Puffer in der freien Liste unter den **minimalen** Wert fällt, versucht der RP, mehr Puffer für diesen Pool zu erstellen.
- **Max-allowed** gibt die maximale Anzahl an Puffern an, die in der freien Liste zulässig sind: Der **max-allowed** Parameter verhindert, dass ein Pool Puffer monopolisiert, die er nicht mehr benötigt. Außerdem wird dieser Speicher für den weiteren Gebrauch wieder in das System zurückgesetzt. Wenn die Anzahl der Puffer in der freien Liste den **maximal zulässigen** Wert übersteigt, sollte der RP versuchen, Puffer aus dem Pool zu entfernen.
- **Treffer** geben die Anzahl der vom Pool angeforderten Puffer an. Der **Treffer-Zähler** bietet

einen Mechanismus, um zu bestimmen, welcher Pool die höchste Nachfrage nach Puffern erfüllen muss.

- **Fehler** geben an, wie oft ein Puffer angefordert und der RP erkannt wurde, in dem der Pool zusätzliche Puffer benötigt wurde. Mit anderen Worten, die Anzahl der Puffer in der freien Liste ist unter die **Min**-Ebene gefallen. Der Zähler **für** Fehlmeldungen stellt die Anzahl der Fälle dar, in denen der RP gezwungen wurde, zusätzliche Puffer zu erstellen.
- **Trims** gibt die Anzahl der Puffer an, die der RP aus dem Pool entfernt hat, wenn die Anzahl der Puffer in der freien Liste die Anzahl der **maximal zulässigen** Puffer überschreitet.
- **Erstellt** die Anzahl der im Pool erstellten Puffer. Der RP erstellt Puffer in folgenden Situationen: Wenn die Nachfrage nach Puffern erhöht ist, bis die Anzahl der Puffer in der freien Liste kleiner als die **Min** Puffer ist. Ein **Fehler** tritt auf, weil es keine Puffer in der freien Liste gibt. Beide vorherigen Situationen.
- **Fehler** identifizieren, wenn IOS keinen kleinen Puffer abrufen, das Paket wird nicht verworfen. Er erhöht den ausgefallenen Zähler und gelangt zum Puffer der nächsten Ebene, dem Mittelpuffer, und fordert dort einen Puffer an. Wenn er keinen mittleren Puffer erhält, wird der Puffer der nächsten Ebene angefordert, der ein großer Puffer ist. Dieser Prozess wird fortgesetzt, bis er den riesigen Puffer-Pool erreicht. Wenn er keinen großen Puffer erhält, verwirft er das Paket.
- **Kein Speicher** identifiziert die Anzahl der Ausfälle, die durch unzureichenden Speicher verursacht werden, um zusätzliche Puffer zu erstellen.

Sie können die Eigenschaften der einzelnen Pools untersuchen, um festzustellen, bei welchen Pools (falls vorhanden) Probleme auftreten. Die Parameter für einen Pool können so angepasst werden, dass der Router besser auf die Auslastung vorbereitet ist, wenn der Pool die folgenden Merkmale aufweist:

- Die Anzahl der Fehlschläge und die Anzahl der Erhöhungen (in Prozent der Treffer).
- In der freien Liste ist die Anzahl der Puffer durchgängig gering.
- Die Anzahl der **Ausfälle** oder **kein Speicher**inkrement.

Konfigurationsbefehl für Puffer

Mit dem Konfigurationsbefehl **puffer** können Sie diese Parameter für jeden Pufferpool anpassen:

- **initial** - Temporäre Puffer, die beim erneuten Laden des Systems zugewiesen werden.
- **max-free**: Maximale Anzahl freier Puffer.
- **min-free**: Mindestanzahl an freien Puffern.
- **Permanent** - Anzahl der permanenten Puffer.

Anfängliche Puffer

Abstimmung der **anfänglichen** Puffer, um den sprunghaften Anstieg des Sitzungsbetriebs nach dem erneuten Laden des Routers zu bewältigen

```
buffers small initial 250
```

Diese Puffer werden schließlich "getrimmt" und wieder an das System zurück.

Die anfänglichen Puffer sind für die Sitzungseinrichtung ausgelegt, die immer prozessorientiert ist.

Während der Sitzungseinrichtung wird der (von anderen Routenprotokollen verwendete) Fast-Switching-Cache aufgefüllt. prozessgesteuerte Puffer sind nicht mehr erforderlich und können an das System zurückgegeben werden.

Die Anpassung der anfänglichen Puffer ist möglicherweise nicht die richtige Lösung für das IBM Feature-Set, da fast alle Pakete (nach der Einrichtung der Sitzung) prozessgesteuert sind und ohnehin zusätzliche Pufferung erfordern.

Hinweis: Für die IBM-prozessgesteuerten Funktionen sollten Sie **permanente** Puffer einstellen, anstatt die temporären ursprünglichen Puffer zu optimieren.

Maximal freie Puffer

Optimieren Sie **max-free**-Puffer, sodass der Wert gleich oder größer als die permanenten Puffer ist. Wenn alle permanenten Puffer in der freien Liste sind, sollte der RP nicht versuchen, permanente Puffer zu trimmen. Mit "Max-free" kann sichergestellt werden, dass nicht verwendete Puffer, die während unregelmäßiger Bursts erstellt werden, an den Systemspeicher zurückgegeben werden.

```
buffers small max-free 175
```

```
buffers small permanent 125
```

Minimalfreie Puffer

Optimieren Sie **minutenfreie** Puffer, sodass der Wert die geschätzte Mindestanzahl der zu jedem Zeitpunkt erforderlichen Puffer darstellt. Min-free kann verwendet werden, um Pufferknappheit vorherzusagen und sicherzustellen, dass immer eine Mindestanzahl an Puffern verfügbar ist.

```
buffers small min-free 50
```

Dauerpuffer

Passen Sie **permanente** Puffer so an, dass der Wert die geschätzte Anzahl der für die normale Verarbeitung erforderlichen Puffer darstellt.

```
buffers small permanent 125
```

Permanente Puffer werden verwendet, um die normalen Pufferanforderungen (einschließlich häufiger Bursts) des Routers zu erfüllen. Die Ermittlung der normalen Pufferanforderungen ist ein interaktiver Prozess, bei dem die **Ausgabe** des **Puffers** die insgesamt in einem Pool verwendeten Puffer zu einem bestimmten Zeitpunkt anzeigen soll. Permanente Puffer sollten auf die konsistenten "Gesamtpuffer" abgestimmt werden. Wenn Sie permanente Puffer einstellen, sollten Sie sich auf die Reduzierung von Erschaffungen und die Beseitigung von Fehlern und Fehlern

konzentrieren.

Zusätzliche Show-Befehle

Es gibt zwei weitere **show**-Befehle, mit denen Sie Probleme bei der Pufferzuweisung identifizieren können:

- **Anzeigen der Schnittstellen-Schnittstellenkennung**
- **show source-bridge**

Diese Beispielausgabe für *den Schnittstellenbezeichner-Schnittstellenbezeichner* enthält einen Zähler für keinen Puffer:

```
dspu-7k#show interfaces channel 4/2
```

```
Channel4/2 is up, line protocol is up
Hardware is cxBus IBM Channel
MTU 4472 bytes, BW 98304 Kbit, DLY 100 usec, rely 255/255, load 1/255
Encapsulation CHANNEL, loopback not set, keepalive not set
Virtual interface
Last input 0:00:04, output 0:00:04, output hang never
Last clearing of "show interface" counters never
Output queue 0/40, 0 drops; input queue 0/75, 8 drops
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
  646 packets input, 27760 bytes, 8 no buffer
  Received 0 broadcasts, 0 runts, 0 giants
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  328 packets output, 16959 bytes, 0 underruns
  0 output errors, 0 collisions, 0 interface resets, 0 restarts
  0 output buffer failures, 0 output buffers swapped out
```

In der **Befehlsausgabe show interfaces interface-identifizier**.

- Der Zähler für `keine Puffer` erhöht sich, wenn die Schnittstelle keinen Puffer für ein eingehendes Paket abruft.
- Sowohl die Zähler `no buffer` als auch die `Drop-Zähler` (Input Queue) werden inkrementiert, wenn die Schnittstelle keinen Puffer für ein eingehendes Paket abruft.
- Ein `kein Pufferzähler`, der in der **Ausgabe der Show-Schnittstellen** inkrementiert, korreliert mit dem `Fehlfunktionszähler`, der in der **Ausgabe der Show-Puffer** inkrementiert ist. Der entsprechende Puffer-Pool kann angepasst werden.

Diese **show source-bridge**-Beispielbefehlsausgabe enthält einen Schnittstellenindikator für Threads, wenn Source-Route Bridging (SRB) für die Schnittstelle konfiguriert ist:

```
dspu-7k#show source-bridge
```

```
Local Interfaces:                receive                transmit
      srn bn  trn r p s n  max hops  cnt:bytes            cnt:bytes            drops
Ch4/2 666  1   99 *  f   7 7 7    652:26020            6:266                0
```

```
Global RSRB Parameters:
```

```
TCP Queue Length maximum: 100
```

```
Ring Group 99:
```

```
This TCP peer: 150.10.20.2
```

```
Maximum output TCP queue length, per peer: 100
```

```

Peers:
      state      bg lv  pkts_rx  pkts_tx  expl_gn  drops  TCP
TCP 150.10.20.1  open          *3    261    266      0      0    0
TCP 150.10.20.2  -            *3      0      0      0      0    0
Rings:
bn: 1  rn: 888  locvrt ma: 4000.7000.fff1 Buff Ring888      fwd: 0
bn: 1  RN: 666  local  ma: 4000.0c48.2e80 Channel4/2      fwd: 261
bn: 1  RN: 88   remote ma: 4000.4000.fff1 TCP 150.10.20.1  fwd: 322
bn: 1  RN: 250  remote ma: 4000.300f.7c09 TCP 150.10.20.1  fwd: 0

Explorers: ----- input -----          ----- output -----
      spanning  all-rings      total      spanning  all-rings      total
Ch4/2          0          0          0          0          1          1

Local: fastswitched 0          flushed 0          max Bps 256000

      rings      inputs      bursts      throttles output drops
Ch4/2          0          0          0          8          0

```

In der Befehlsausgabe **show source-bridge**:

- Der Throttles-Zähler erhöht sich, wenn die Schnittstelle keinen Puffer für ein eingehendes Paket abrufen kann.
- Der `Throttles`-Zähler, der in der Ausgabe des Befehls **show interfaces** inkrementiert, korreliert mit einem `Fehlfunktionszähler`, der in der Befehlsausgabe des Befehls **show buffers** inkrementiert. Der entsprechende Puffer-Pool kann angepasst werden.

[Zugehörige Informationen](#)

- [Puffer-Tuning für alle Cisco Router](#)
- [Technischer Support und Dokumentation - Cisco Systems](#)