

# Dépannage des fuites de mémoire tampon

## Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Components Used](#)

[Conventions](#)

[Fuites de mémoire tampon d'interface pondérées](#)

[Fuites de mémoire tampon système](#)

[Conseils de dépannage](#)

[Informations connexes](#)

## Introduction

Les fuites de mémoire tampon sont des bogues logiciels connus de Cisco IOSMD. Il existe deux types de fuites de mémoire tampon :

- Fuites de tampon d'interface pondérées.
- Fuites de mémoire tampon système.

Afin de dépanner les fuites de mémoire tampon, vous devez identifier le type de fuite de mémoire tampon que vous rencontrez. Les commandes **show interfaces** et **show buffers** sont très utiles dans cette situation.

Si vous disposez des résultats des commandes **show interfaces** et **show buffers** de votre périphérique Cisco, vous pouvez utiliser [Cisco CLI Analyzer](#) pour afficher les problèmes potentiels et les correctifs. Pour utiliser [Cisco CLI Analyzer](#), vous devez être un client [inscrit, être connecté et avoir activé JavaScript](#).

## Conditions préalables

### Conditions requises

Aucune spécification déterminée n'est requise pour ce document.

### Components Used

Ce document n'est pas limité à des versions de matériel et de logiciel spécifiques.

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. If your network is live, make sure that you understand the potential impact of any command.

### Conventions

For more information on document conventions, refer to the [Cisco Technical Tips Conventions](#).

## Fuites de mémoire tampon d'interface pondérées

Les fuites de tampon d'interface pondérées entraînent le remplissage de la file d'attente d'entrée d'une interface jusqu'au point où elle ne peut plus accepter de paquets. Dans certaines conditions de trafic spécifiques, la file d'attente d'entrée sur une interface devient coincée ou, en d'autres termes, le nombre de files d'attente d'entrée est supérieur à la profondeur de la file d'attente.

Voici un exemple de sortie de la commande **show interfaces**, qui montre que l'interface est coincée :

```
Ethernet0/0 is up, line protocol is up  
Output queue 0/40, 0 drops; input queue 76/75, 1250 drops
```

Le symptôme d'une telle fuite de tampon est une file d'attente d'entrée complète (76/75). Ici, les valeurs 76 et 75 représentent respectivement le nombre de paquets dans la file d'attente d'entrée et la taille maximale de la file d'attente d'entrée. Dans ce cas, le nombre de paquets dans la file d'attente d'entrée est supérieur à la profondeur de la file d'attente. C'est ce qu'on appelle une « interface coincée ». Lorsqu'une interface est coincée, le routeur ne transfère plus le trafic provenant de l'interface affectée.

Rechargez le routeur pour libérer la file d'attente d'entrée et restaurer le trafic jusqu'à ce que la file d'attente soit à nouveau pleine. Cela peut prendre entre quelques secondes et quelques semaines, en fonction de la gravité de la fuite.

**Attention** : Avant de recharger le routeur, assurez-vous de collecter toutes les informations nécessaires pour identifier le coupable.

Utilisez ces commandes pour identifier la source de la fuite de mémoire tampon :

- **show buffers pool [nom du pool] [paquet/en-tête]**
- **show buffers old** (Utilisez cette commande uniquement si **debug sanity** est activé. **Remarque** : La commande **debug sanity** est masquée dans la plupart des versions du logiciel Cisco IOS. Lorsque **debug sanity** est activé, chaque tampon utilisé dans le système est vérifié lors de son allocation, et à nouveau lors de sa libération. **Remarque** : Vous devez exécuter la commande **debug sanity** en mode d'exécution privilégié (mode enable). Bien que cette commande utilise une certaine capacité de CPU, elle n'affecte pas de manière significative les fonctionnalités du routeur. Comme les autres commandes debug, **debug sanity** n'est pas enregistré dans la configuration. Par conséquent, cette commande ne survivra pas à un redémarrage du système. **Remarque** : Afin de désactiver le contrôle de la santé mentale, utilisez la commande EXEC privilégiée **undebug sanity**.)
- **show buffer assigné**

## Fuites de mémoire tampon système

Cette section traite des fuites de mémoire tampon système.

Voici un exemple de sortie de la commande **show buffers**, qui indique une fuite de tampon dans l'un des pools de tampons système :

```
Middle buffers, 600 bytes (total 20825, permanent 180):
 286 in free list (20 min, 400 max allowed)
 89122311 hits, 99597 misses, 133679 trims, 154324 created
 2247 failures (0 no memory)
```

Cette sortie de commande **show buffers** indique une fuite de mémoire tampon dans le pool de mémoires tampon intermédiaires. Il y a un total de 20825 mémoires tampon intermédiaires dans le routeur, et seulement 286 sont dans la liste libre. Cela implique que certains processus prennent toutes les tampons, mais ne les retournent pas.

Les autres symptômes de ce type de fuite de mémoire tampon sont les messages d'erreur "%SYS-2-MALLOCFAIL » pour le processeur de pool ou les entrées/sorties (E/S), basés sur la plate-forme.

Utilisez ces commandes pour identifier la source de la fuite de mémoire tampon :

- **show buffers old** (Utilisez cette commande uniquement si **debug sanity** est activé. **Remarque** : La commande **debug sanity** est masquée dans la plupart des versions du logiciel Cisco IOS. Lorsque **debug sanity** est activé, chaque tampon utilisé dans le système est vérifié lors de son allocation, et à nouveau lors de sa libération. **Remarque** : Vous devez exécuter la commande **debug sanity** en mode d'exécution privilégié (mode enable). Bien que cette commande utilise une certaine capacité de CPU, elle n'affecte pas de manière significative les fonctionnalités du routeur. Comme les autres commandes debug, **debug sanity** n'est pas enregistré dans la configuration. Par conséquent, cette commande ne survivra pas à un redémarrage du système. **Remarque** : Afin de désactiver le contrôle de la santé mentale, utilisez la commande EXEC privilégiée **undebug sanity**.)
- **show buffers pool [nom du pool] [paquet/en-tête]**
- **show buffer assigné**

## Conseils de dépannage

Les fuites de mémoire tampon sont des bogues du logiciel Cisco IOS. Afin de corriger les bogues connus de fuite de tampon, mettez à niveau vers la dernière version de votre groupe de versions. Par exemple, si vous exécutez actuellement le logiciel Cisco IOS Version 11.2(14), mettez à niveau vers la dernière image 11.2(x). Si cela n'aide pas ou s'il n'est pas possible de mettre à niveau le routeur, contactez le centre d'assistance technique de Cisco et fournissez à l'ingénieur le résultat des commandes **show buffers** appropriées, ainsi que le résultat de la commande **show tech-support**.

Voici quelques conseils pour vous aider à identifier les paquets qui provoquent une fuite de mémoire tampon :

- Lorsque vous détectez une fuite de mémoire tampon, utilisez les commandes **show buffers** associées pour trouver un modèle dans les paquets qui utilisent tant de mémoires tampon.
- Lorsque vous identifiez le type de paquets, essayez de trouver une solution pour empêcher la fuite (par exemple, utilisez une liste d'accès pour filtrer ces paquets).

Voici des exemples de sortie des commandes **show** associées :

```
Router#show interface ethernet 0/0
Ethernet0/0 is up, line protocol is up
```

```

Hardware is AmdP2, address is 0050.3ee8.4060 (bia 0050.3ee8.4060)
Internet address is 10.200.40.37/22
MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec, rely 255/255, load 1/255
Encapsulation ARPA, loopback not set, keepalive set (10 sec)
ARP type: ARPA, ARP Timeout 04:00:00
Last input 00:00:51, output 00:00:08, output hang never
Last clearing of "show interface" counters never
Queueing strategy: fifo
Output queue 0/40, 0 drops; input queue 76/75, 1250 drops
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
  15686 packets input, 2872866 bytes, 0 no buffer
  Received 15342 broadcasts, 0 runts, 0 giants, 0 throttles
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input packets with dribble condition detected
  10352 packets output, 1031158 bytes, 0 underruns
  0 output errors, 0 collisions, 3 interface resets
  0 babbles, 0 late collision, 2 deferred
  0 lost carrier, 0 no carrier
  0 output buffer failures, 0 output buffers swapped out

```

Router#**show buffers old**

Header	DataArea	Pool	Rcnt	Size	Link	Enc	Flags	Input	Output
80F09828	1A00084	Small	1	54	11	11	201	Et0/0	None
80F09A34	1A001C4	Small	1	54	11	11	201	Et0/0	None
80F09C40	1A00304	Small	1	54	11	11	201	Et0/0	None
80F09E4C	1A00444	Small	1	54	11	11	201	Et0/0	None
80F0A058	1A00584	Small	1	54	11	11	201	Et0/0	None
80F0A264	1A006C4	Small	1	54	11	11	201	Et0/0	None
80F0A470	1A00804	Small	1	54	11	11	201	Et0/0	None
80F0A67C	1A00944	Small	1	54	11	11	201	Et0/0	None
80F0A888	1A00A84	Small	1	54	11	11	201	Et0/0	None
80F0AA94	1A00BC4	Small	1	54	11	11	201	Et0/0	None
80F0ACA0	1A00D04	Small	1	54	11	11	201	Et0/0	None
80F0AEAC	1A00E44	Small	1	54	11	11	201	Et0/0	None
80F0B0B8	1A00F84	Small	1	54	11	11	201	Et0/0	None
80F0B2C4	1A010C4	Small	1	54	11	11	201	Et0/0	None
80F0B4D0	1A01204	Small	1	54	11	11	201	Et0/0	None
80F0B6DC	1A01344	Small	1	54	11	11	201	Et0/0	None
80F0B8E8	1A01484	Small	1	54	11	11	201	Et0/0	None
80F0BAF4	1A015C4	Small	1	54	11	11	201	Et0/0	None
80F0BD00	1A01704	Small	1	54	11	11	201	Et0/0	None
80F0BF0C	1A01844	Small	1	54	11	11	201	Et0/0	None
80F0C118	1A01984	Small	1	54	11	11	201	Et0/0	None
80F0C324	1A01AC4	Small	1	54	11	11	201	Et0/0	None
80F0C530	1A01C04	Small	1	54	11	11	201	Et0/0	None
80F0C73C	1A01D44	Small	1	54	11	11	201	Et0/0	None
80F5F644	1B9B0A4	Small	1	54	11	11	201	Et0/0	None
80FDF118	1B78604	Small	1	54	11	11	201	Et0/0	None
80FDF324	1B78744	Small	1	54	11	11	201	Et0/0	None
80FDF530	1B78884	Small	1	54	11	11	201	Et0/0	None
80FDF73C	1B789C4	Small	1	54	11	11	201	Et0/0	None
80FDF948	1B78B04	Small	1	54	11	11	201	Et0/0	None
80FDFB54	1B78C44	Small	1	54	11	11	201	Et0/0	None
80FDFD60	1B78D84	Small	1	54	11	11	201	Et0/0	None
80FDFE6C	1B78EC4	Small	1	54	11	11	201	Et0/0	None
80FE0178	1B79004	Small	1	54	11	11	201	Et0/0	None
80FE0384	1B79144	Small	1	54	11	11	201	Et0/0	None
80FE0590	1B79284	Small	1	54	11	11	201	Et0/0	None
80FE079C	1B793C4	Small	1	54	11	11	201	Et0/0	None
80FE09A8	1B79504	Small	1	54	11	11	201	Et0/0	None
80FE0BB4	1B79644	Small	1	54	11	11	201	Et0/0	None

80FE0DC0	1B79784	Small	1	54	11	11	201	Et0/0	None
80FE0FCC	1B798C4	Small	1	54	11	11	201	Et0/0	None
80FE11D8	1B79A04	Small	1	54	11	11	201	Et0/0	None
80FE13E4	1B79B44	Small	1	54	11	11	201	Et0/0	None
80FE15F0	1B79C84	Small	1	54	11	11	201	Et0/0	None
80FE17FC	1B79DC4	Small	1	54	11	11	201	Et0/0	None
80FE1A08	1B79F04	Small	1	54	11	11	201	Et0/0	None
80FE1C14	1B7A044	Small	1	54	11	11	201	Et0/0	None
80FE1E20	1B7A184	Small	1	54	11	11	201	Et0/0	None
80FE202C	1B7A2C4	Small	1	54	11	11	201	Et0/0	None
80FE2238	1B7A404	Small	1	54	11	11	201	Et0/0	None
81107F40	1B9B1E4	Small	1	54	11	11	201	Et0/0	None
8110814C	1B9B324	Small	1	54	11	11	201	Et0/0	None
81108358	1B9B464	Small	1	54	11	11	201	Et0/0	None
81108564	1B9B5A4	Small	1	54	11	11	201	Et0/0	None
8110897C	1B9B824	Small	1	54	11	11	201	Et0/0	None
81108B88	1B9B964	Small	1	54	11	11	201	Et0/0	None
81108D94	1B9BAA4	Small	1	54	11	11	201	Et0/0	None
81108FA0	1B9BBE4	Small	1	54	11	11	201	Et0/0	None
811093B8	1B9BE64	Small	1	54	11	11	201	Et0/0	None
811095C4	1B9BFA4	Small	1	54	11	11	201	Et0/0	None
811097D0	1B9C0E4	Small	1	54	11	11	201	Et0/0	None
811099DC	1B9C224	Small	1	54	11	11	201	Et0/0	None
81109DF4	1B9C4A4	Small	1	54	11	11	201	Et0/0	None
8110A000	1B9C5E4	Small	1	54	11	11	201	Et0/0	None
8110A20C	1B9C724	Small	1	54	11	11	201	Et0/0	None
8110A418	1B9C864	Small	1	54	11	11	201	Et0/0	None
81121364	1B9CC24	Small	1	54	11	11	201	Et0/0	None
81121570	1B9CD64	Small	1	54	11	11	201	Et0/0	None
81121988	1B9CFE4	Small	1	54	11	11	201	Et0/0	None
81121B94	1B9D124	Small	1	54	11	11	201	Et0/0	None
81121FAC	1B9D3A4	Small	1	54	11	11	201	Et0/0	None
811221B8	1B9D4E4	Small	1	54	11	11	201	Et0/0	None
811225D0	1B9D764	Small	1	54	11	11	201	Et0/0	None
811227DC	1B9D8A4	Small	1	54	11	11	201	Et0/0	None
811229E8	1B9D9E4	Small	1	54	11	11	201	Et0/0	None
81122BF4	1B9DB24	Small	1	54	11	11	201	Et0/0	None

Router#**show buffers old header**

Buffer information for Small buffer at 0x80F09828

```

data_area 0x1A00084, refcount 1, next 0x0, flags 0x201
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
if_input 0x80F57BE0 (Ethernet0/0), if_output 0x0 (None)
inputtime 0x4CDFC58, outputtime 0x0, oqnumber 65535
datagramstart 0x1A000CA, datagramsize 54, maximum size 260
mac_start 0x1A000CA, addr_start 0x1A000CA, info_start 0x0
network_start 0x1A000D8, transport_start 0x0

```

```

source:BE200040.0060.09c3.f9fe socket 0453
destination: BE200040.ffff.ffff.ffff socket 0453 protocol 01

```

Buffer information for Small buffer at 0x80F09A34

```

data_area 0x1A001C4, refcount 1, next 0x0, flags 0x201
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
if_input 0x80F57BE0 (Ethernet0/0), if_output 0x0 (None)
inputtime 0x4CDFAA0, outputtime 0x0, oqnumber 65535
datagramstart 0x1A0020A, datagramsize 54, maximum size 260
mac_start 0x1A0020A, addr_start 0x1A0020A, info_start 0x0
network_start 0x1A00218, transport_start 0x0

```

```

source:BE200040.0060.09c3.f9fe socket 0453
destination: BE200040.ffff.ffff.ffff socket 0453 protocol 01

```

```

Buffer information for Small buffer at 0x80F09C40
  data_area 0x1A00304, refcount 1, next 0x0, flags 0x201
  linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
  if_input 0x80F57BE0 (Ethernet0/0), if_output 0x0 (None)
  inputtime 0x4CDF8D7, outputtime 0x0, oqnumber 65535
  datagramstart 0x1A0034A, datagramsize 54, maximum size 260
  mac_start 0x1A0034A, addr_start 0x1A0034A, info_start 0x0
  network_start 0x1A00358, transport_start 0x0

```

```

source:BE200040.0060.09c3.f9fe socket 0453
destination: BE200040.ffff.ffff.ffff socket 0453 protocol 01

```

....

```
Router#show buffers input-interface ethernet 0/0
```

Header	DataArea	Pool	Rcnt	Size	Link	Enc	Flags	Input	Output
80F09828	1A00084	Small	1	54	11	11	201	Et0/0	None
80F09A34	1A001C4	Small	1	54	11	11	201	Et0/0	None
80F09C40	1A00304	Small	1	54	11	11	201	Et0/0	None
80F09E4C	1A00444	Small	1	54	11	11	201	Et0/0	None
80F0A058	1A00584	Small	1	54	11	11	201	Et0/0	None
80F0A264	1A006C4	Small	1	54	11	11	201	Et0/0	None
80F0A470	1A00804	Small	1	54	11	11	201	Et0/0	None
80F0A67C	1A00944	Small	1	54	11	11	201	Et0/0	None
80F0A888	1A00A84	Small	1	54	11	11	201	Et0/0	None
80F0AA94	1A00BC4	Small	1	54	11	11	201	Et0/0	None
80F0ACA0	1A00D04	Small	1	54	11	11	201	Et0/0	None
80F0AEAC	1A00E44	Small	1	54	11	11	201	Et0/0	None
80F0B0B8	1A00F84	Small	1	54	11	11	201	Et0/0	None
80F0B2C4	1A010C4	Small	1	54	11	11	201	Et0/0	None
80F0B4D0	1A01204	Small	1	54	11	11	201	Et0/0	None
80F0B6DC	1A01344	Small	1	54	11	11	201	Et0/0	None
80F0B8E8	1A01484	Small	1	54	11	11	201	Et0/0	None
80F0BAF4	1A015C4	Small	1	54	11	11	201	Et0/0	None
80F0BD00	1A01704	Small	1	54	11	11	201	Et0/0	None
80F0BF0C	1A01844	Small	1	54	11	11	201	Et0/0	None
80F0C118	1A01984	Small	1	54	11	11	201	Et0/0	None
80F0C324	1A01AC4	Small	1	54	11	11	201	Et0/0	None
80F0C530	1A01C04	Small	1	54	11	11	201	Et0/0	None
80F0C73C	1A01D44	Small	1	54	11	11	201	Et0/0	None
80F5F644	1B9B0A4	Small	1	54	11	11	201	Et0/0	None
80FDF118	1B78604	Small	1	54	11	11	201	Et0/0	None
80FDF324	1B78744	Small	1	54	11	11	201	Et0/0	None
80FDF530	1B78884	Small	1	54	11	11	201	Et0/0	None
80FDF73C	1B789C4	Small	1	54	11	11	201	Et0/0	None
80FDF948	1B78B04	Small	1	54	11	11	201	Et0/0	None
80FDFB54	1B78C44	Small	1	54	11	11	201	Et0/0	None
80FDFD60	1B78D84	Small	1	54	11	11	201	Et0/0	None
80FDFE6C	1B78EC4	Small	1	54	11	11	201	Et0/0	None
80FE0178	1B79004	Small	1	54	11	11	201	Et0/0	None
80FE0384	1B79144	Small	1	54	11	11	201	Et0/0	None
80FE0590	1B79284	Small	1	54	11	11	201	Et0/0	None
80FE079C	1B793C4	Small	1	54	11	11	201	Et0/0	None
80FE09A8	1B79504	Small	1	54	11	11	201	Et0/0	None
80FE0BB4	1B79644	Small	1	54	11	11	201	Et0/0	None
80FE0DC0	1B79784	Small	1	54	11	11	201	Et0/0	None
80FE0FCC	1B798C4	Small	1	54	11	11	201	Et0/0	None
80FE11D8	1B79A04	Small	1	54	11	11	201	Et0/0	None
80FE13E4	1B79B44	Small	1	54	11	11	201	Et0/0	None
80FE15F0	1B79C84	Small	1	54	11	11	201	Et0/0	None
80FE17FC	1B79DC4	Small	1	54	11	11	201	Et0/0	None
80FE1A08	1B79F04	Small	1	54	11	11	201	Et0/0	None

80FE1C14	1B7A044	Small	1	54	11	11	201	Et0/0	None
80FE1E20	1B7A184	Small	1	54	11	11	201	Et0/0	None
80FE202C	1B7A2C4	Small	1	54	11	11	201	Et0/0	None
80FE2238	1B7A404	Small	1	54	11	11	201	Et0/0	None
81107F40	1B9B1E4	Small	1	54	11	11	201	Et0/0	None
8110814C	1B9B324	Small	1	54	11	11	201	Et0/0	None
81108358	1B9B464	Small	1	54	11	11	201	Et0/0	None
81108564	1B9B5A4	Small	1	54	11	11	201	Et0/0	None
8110897C	1B9B824	Small	1	54	11	11	201	Et0/0	None
81108B88	1B9B964	Small	1	54	11	11	201	Et0/0	None
81108D94	1B9BAA4	Small	1	54	11	11	201	Et0/0	None
81108FA0	1B9BBE4	Small	1	54	11	11	201	Et0/0	None
811093B8	1B9BE64	Small	1	54	11	11	201	Et0/0	None
811095C4	1B9BFA4	Small	1	54	11	11	201	Et0/0	None
811097D0	1B9C0E4	Small	1	54	11	11	201	Et0/0	None
811099DC	1B9C224	Small	1	54	11	11	201	Et0/0	None
81109DF4	1B9C4A4	Small	1	54	11	11	201	Et0/0	None
8110A000	1B9C5E4	Small	1	54	11	11	201	Et0/0	None
8110A20C	1B9C724	Small	1	54	11	11	201	Et0/0	None
8110A418	1B9C864	Small	1	54	11	11	201	Et0/0	None
81121364	1B9CC24	Small	1	54	11	11	201	Et0/0	None
81121570	1B9CD64	Small	1	54	11	11	201	Et0/0	None
81121988	1B9CFE4	Small	1	54	11	11	201	Et0/0	None
81121B94	1B9D124	Small	1	54	11	11	201	Et0/0	None
81121FAC	1B9D3A4	Small	1	54	11	11	201	Et0/0	None
811221B8	1B9D4E4	Small	1	54	11	11	201	Et0/0	None
811225D0	1B9D764	Small	1	54	11	11	201	Et0/0	None
811227DC	1B9D8A4	Small	1	54	11	11	201	Et0/0	None
811229E8	1B9D9E4	Small	1	54	11	11	201	Et0/0	None
81122BF4	1B9DB24	Small	1	54	11	11	201	Et0/0	None

Router#show buffers address 81122BF4 dump

```

Buffer information for Small buffer at 0x81122BF4
  data_area 0x1B9DB24, refcount 1, next 0x0, flags 0x201
  linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
  if_input 0x80F57BE0 (Ethernet0/0), if_output 0x0 (None)
  inputtime 0x4CE2BFC, outputtime 0x0, oqnumber 65535
  datagramstart 0x1B9DB6A, datagramsize 54, maximum size 260
  mac_start 0x1B9DB6A, addr_start 0x1B9DB6A, info_start 0x0
  network_start 0x1B9DB78, transport_start 0x0

source:BE200040.0060.09c3.f9fe socket 0453
destination: BE200040.ffff.ffff.ffff socket 0453 protocol 01

```

```

01B9DB20: 00000000 00000000 00000000 00000000 .....
01B9DB30: 00000000 00000000 00000000 00000000 .....
01B9DB40: 00000000 00000000 00000000 00000000 .....
01B9DB50: 00000000 00000000 00000000 00000000 .....
01B9DB60: 00000000 00000000 0000FFFF FFFFFFFF .....
01B9DB70: 006009C3 F9FE0028 FFFF0028 0001BE20 .`.Cy~.(...(.>
01B9DB80: 0040FFFF FFFFFFFF 0453BE20 00400060 .@.....S> .@.`
01B9DB90: 09C3F9FE 04530001 00000040 06000200 .Cy~.S.....@....
01B9DBA0: 00000000 00000000 00000000 00000000 .....
01B9DBB0: 00000000 00000000 00000000 00000000 .....
01B9DBC0: 00000000 00000000 00000000 00000000 .....
01B9DBD0: 00000000 00000000 00000000 00000000 .....
01B9DBE0: 00000000 00000000 00000000 00000000 .....
01B9DBF0: 00000000 00000000 00000000 00000000 .....
01B9DC00: 00000000 00000000 00000000 00000000 .....
01B9DC10: 00000000 00000000 00000000 00000000 .....
01B9DC20: 00000000 00 .....

```

Router#

Si vous ne parvenez pas à identifier un modèle dans les mémoires tampon, capturez le résultat des commandes **show** (par exemple, **show buffers old**) et enregistrez-le dans un fichier (par exemple, buffers.log). Ensuite, essayez d'isoler le motif à l'aide de l'utilitaire « **grep** » UNIX, ou quelque chose de similaire.

```
grep linktype buffers.log
```

```
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 0 (None), enctype 0 (None), encsize 0, rxtype 0
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
```

...

*!--- Here you can clearly see a lot of NOVELL-related buffers.*

*!--- The problem seems to be with the IPX packets.*

*!--- You can check this through the **wc -l** (to count lines) command on a UNIX system.*

```
grep linktype buffers.log | wc -l
```

```
175
```

```
grep linktype buffers.log | grep NOVELL-ETHER | wc -l
```

```
153
```

*!--- 153 out of 175 old buffers are IPX packets. Try to find out what*

*!--- type of packets they are with another **grep** command:*

```
grep socket buffers.log
```

```
source:BE200040.0060.09c3.f9fe socket 0453
destination: BE200040.ffff.ffff.ffff socket 0453 protocol 01
source:BE200040.0060.09c3.f9fe socket 0453
destination: BE200040.ffff.ffff.ffff socket 0453 protocol 01
source:BE200040.0060.09c3.f9fe socket 0453
destination: BE200040.ffff.ffff.ffff socket 0453 protocol 01
source:BE200040.0060.09c3.f9fe socket 0453
destination: BE200040.ffff.ffff.ffff socket 0453 protocol 01
source:BE200040.0060.09c3.f9fe socket 0453
destination: BE200040.ffff.ffff.ffff socket 0453 protocol 01
source:BE200040.0060.09c3.f9fe socket 0453
destination: BE200040.ffff.ffff.ffff socket 0453 protocol 01
source:BE200040.0060.09c3.f9fe socket 0453
destination: BE200040.ffff.ffff.ffff socket 0453 protocol 01
source:BE200040.0060.09c3.f9fe socket 0453
destination: BE200040.ffff.ffff.ffff socket 0453 protocol 01
source:BE200040.0060.09c3.f9fe socket 0453
```

...

*!--- There are Broadcasts to socket 453, protocol 01...*

*!--- Those are IPX RIP packets.*

*!--- Disable IPX RIP, or use IPX EIGRP instead, until a bug fix is available.*

En résumé :



- Vérifiez si vous avez une fuite de mémoire tampon. Les fuites de mémoire tampon sont souvent mal interprétées comme une rafale de trafic (avec de nombreux paquets qui passent à la commutation de processus en raison d'une configuration incorrecte ou d'une fonctionnalité non prise en charge), ou comme une attaque.
- Les fuites de mémoire tampon sont des bogues du logiciel Cisco IOS. La meilleure solution pour ce problème est de mettre à niveau le logiciel Cisco IOS vers la dernière version.
- Si cela échoue, contactez le centre d'assistance technique Cisco et fournissez à l'ingénieur les résultats des commandes **show buffers** et **show tech-support** appropriées.

## Informations connexes

- [Réglage du tampon](#)
- [Dépannage des problèmes de mémoire](#)
- [Support technique - Cisco Systems](#)