

# Resolución de problemas de fuga de memoria intermedia

## Contenido

[Introducción](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Convenciones](#)

[Fugas de memoria intermedia de la interfaz con cuña](#)

[Fugas de memoria intermedia del sistema](#)

[Sugerencias para Resolver Problemas](#)

[Información Relacionada](#)

## Introducción

Las fugas de memoria intermedia son los errores de funcionamiento del software Cisco IOS®. Hay dos tipos diferentes de fugas en memoria intermedia:

- Pérdidas acunadas del búfer de interfaz.
- Fugas de memoria intermedia del sistema.

Para resolver problemas de fugas de memoria intermedia, debe identificar el tipo de fuga de memoria intermedia que encuentra. Los comandos `show interfaces` y `show buffers` son muy útiles en esta situación.

Si tiene el resultado de los comandos `show interfaces` y `show buffers` de su dispositivo Cisco, puede utilizar [Cisco CLI Analyzer](#) para mostrar posibles problemas y soluciones. Para utilizar la herramienta [Analizador Cisco CLI](#), debe ser un [cliente registrado, iniciar sesión y tener JavaScript habilitado](#).

## Prerequisites

### Requirements

No hay requisitos específicos para este documento.

### Componentes Utilizados

Este documento no tiene restricciones específicas en cuanto a versiones de software y de hardware.

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.

## Convenciones

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

## Fugas de memoria intermedia de la interfaz con cuña

Las fugas de memoria intermedia de interfaz boqueadas hacen que la cola de entrada de una interfaz se llene hasta el punto en que ya no puede aceptar paquetes. En algunas condiciones de tráfico específicas, la cola de entrada en una interfaz se atasca o, en otras palabras, el conteo de cola de entrada es mayor que la profundidad de la cola.

A continuación se muestra un ejemplo de salida del comando **show interfaces**, que muestra que la interfaz está atascada:

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

El síntoma de tal fuga de memoria intermedia es una cola de entrada completa (76/75). Aquí, los valores 76 y 75 representan el número de paquetes en la cola de entrada y el tamaño máximo de la cola de entrada, respectivamente. En este caso, el número de paquetes en la cola de entrada es mayor que la profundidad de la cola. Esto se denomina “interfaz en cuña”. Cuando una interfaz está atada, el router ya no reenvía el tráfico que viene de la interfaz afectada.

Recargue el router para liberar la cola de entrada y restaurar el tráfico hasta que la cola esté llena otra vez. Esto puede tardar entre unos segundos y unas semanas, en función de la gravedad de la filtración.

**Precaución:** Antes de recargar el router, asegúrese de recopilar toda la información necesaria para identificar al culpable.

Utilice estos comandos para identificar el origen de la fuga del búfer:

- **show buffers pool [pool name] [packet/header]**
- **show buffers old** (Utilice este comando sólo si **debug sanity** está habilitado. **Nota:** El comando **debug sanity** se oculta en la mayoría de las versiones de software del IOS de Cisco. Cuando el comando **debug sanity** está habilitado, se comprueba la integridad de cada búfer empleado en el sistema tanto al asignarse como al liberarse. **Nota:** Debe ejecutar el comando **debug sanity** en modo EXEC privilegiado (modo enable). Aunque este comando utiliza cierta capacidad de CPU, no afecta significativamente la funcionalidad del router. Al igual que otros comandos **debug**, **debug sanity** no se guarda en la configuración. Por lo tanto, este comando no sobrevivirá a un reinicio del sistema. **Nota:** Para inhabilitar la verificación de integridad, utilice el comando EXEC privilegiado **undebug sanity**.)
- **show buffer assigned**

## Fugas de memoria intermedia del sistema

Esta sección hace referencia a fugas del búfer del sistema.

A continuación se muestra un ejemplo de salida del comando **show buffers**, que indica una fuga

de memoria intermedia en uno de los agrupamientos de memoria intermedia del sistema:

```
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)
```

Esta salida del comando **show buffers** indica una fuga de memoria intermedia en el conjunto de memorias intermedias. Existe un total de 20825 memorias intermedias medianas en el router y sólo 286 en la lista disponible. Esto implica que algún proceso toma todas las memorias intermedias, pero no las devuelve.

Otros síntomas de este tipo de fuga de búfer son los mensajes de error "%SYS-2-MALLOCFAIL" para el procesador del grupo o la entrada/salida (E/S), según la plataforma.

Utilice estos comandos para identificar el origen de la fuga del búfer:

- **show buffers old** (Utilice este comando sólo si **debug sanity** está habilitado. **Nota:** El comando **debug sanity** se oculta en la mayoría de las versiones de software del IOS de Cisco. Cuando el comando **debug sanity** está habilitado, se comprueba la integridad de cada búfer empleado en el sistema tanto al asignarse como al liberarse. **Nota:** Debe ejecutar el comando **debug sanity** en modo EXEC privilegiado (modo enable). Aunque este comando utiliza cierta capacidad de CPU, no afecta significativamente la funcionalidad del router. Al igual que otros comandos **debug**, **debug sanity** no se guarda en la configuración. Por lo tanto, este comando no sobrevivirá a un reinicio del sistema. **Nota:** Para inhabilitar la verificación de integridad, utilice el comando EXEC privilegiado **undebg sanity**.)
- **show buffers pool [pool name] [packet/header]**
- **show buffer assigned**

## Sugerencias para Resolver Problemas

Las fugas de memoria intermedia son errores de software del IOS de Cisco. Para corregir los errores conocidos de fuga de memoria intermedia, actualice a la última versión en su tren de versión. Por ejemplo, si actualmente ejecuta Cisco IOS Software Release 11.2(14), actualice a la última imagen 11.2(x). Si esto no ayuda, o si no es posible actualizar el router, comuníquese con el TAC de Cisco y proporcione al ingeniero la salida de los comandos **show buffers** relevantes, y la salida del comando **show tech-support**.

Estos son algunos consejos para ayudarle a identificar los paquetes que causan la fuga del búfer:

- Cuando detecta una fuga de memoria intermedia, utilice los comandos **show buffers** asociados para encontrar un patrón en los paquetes que utilizan tantas memorias intermedias.
- Cuando identifique el tipo de paquetes, intente encontrar una solución para evitar la fuga (por ejemplo, utilice una lista de acceso para filtrar esos paquetes).

Aquí hay ejemplos de salida de comandos **show** asociados:

```
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 no puede identificar un patrón en las memorias intermedias, capture el resultado de los comandos **show** (por ejemplo, **show buffers old**) y guárdelo en un archivo (como buffers.log). Luego, intente aislar el patrón con la ayuda de la utilidad UNIX "grep", o algo similar.

```
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 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
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
destination: BE200040.ffff.ffff.ffff socket 0453 protocol 01
...
!--- 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 resumen:



- Verifique si tiene una pérdida de memoria intermedia. Las fugas de memoria intermedia a menudo se malinterpretan como una ráfaga de tráfico (con muchos paquetes que van a process-switching debido a una configuración incorrecta o a una función no admitida), o como un ataque.
- Las fugas de memoria intermedia son errores de software del IOS de Cisco. La mejor solución para este problema es actualizar el software Cisco IOS a la última versión.
- Si esto falla, póngase en contacto con el TAC de Cisco y proporcione al ingeniero la salida de los comandos **show buffers** y **show tech-support**.

## Información Relacionada

- [Ajuste de la memoria intermedia](#)
- [Resolución de problemas de la memoria](#)
- [Soporte Técnico - Cisco Systems](#)