

75xx/76xx Router: Configuración y verificación de las funciones distribuidas

Contenido

[Introducción](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Convenciones](#)

[Funciones distribuidas](#)

[MLPPP distribuido](#)

[LFI distribuido](#)

[Diferencias entre dMLP y dLFloLL](#)

[MLFR distribuido](#)

[DDR distribuido](#)

[Restricciones y requisitos previos de las funciones distribuidas](#)

[Número de paquetes, enlaces y requisitos de memoria](#)

[Hardware y software MLPPP o MLFR en tarjetas de línea SIP 7600](#)

[Vida de un paquete](#)

[Ruta De Datos Rx](#)

[Ruta de datos Tx](#)

[Reensamblado](#)

[Configuración, Verificación y Depuración de Funciones Distribuidas](#)

[Configuración y Verificación de dMFR](#)

[Configuración y Verificación de dMLP/dLFloLL](#)

[Configuración y Verificación de dLFloFR y dLFloATM](#)

[Configuración y Verificación de dDDR](#)

[Depuración dMLP y dDDR](#)

[Preguntas Frecuentes](#)

[Mejoras de depuración](#)

[Información Relacionada](#)

[Introducción](#)

Este documento le ayuda a entender, configurar y verificar estas características:

- Protocolo punto a punto multilink distribuido (dMLP)
- Fragmentación y entrelazado de enlaces distribuidos (LFI)
- LFI distribuido sobre línea arrendada (dLFloLL)
- LFI distribuido sobre Frame Relay (dLFloFR)

- LFI distribuido sobre ATM (dLFloATM)
- Diferencias entre MLP distribuido (dMLP) y dLFloLL
- Frame Relay de links múltiples distribuidos (dMLFR)
- Routing de marcación a petición distribuido (DDR)

Prerequisites

Requirements

Los lectores de este documento deben tener conocimiento de las funciones distribuidas para Cisco 7500/7600/6500.

Componentes Utilizados

La información que contiene este documento se basa en las siguientes versiones de software y hardware.

- Todas las plataformas 7500 y 7600 de Cisco **Nota:** La información de este documento también se aplica a las plataformas 6500.
- Versiones relevantes del software Cisco IOS®, que en esta tabla se enumeran:

Compatibilidad con funciones distribuidas para cada sucursal y plataforma

Función	Adaptadores de puerto (PA) ¹ admitidos	7500 plataformas		7600 plataformas	
		Versiones principales del software Cisco IOS	Versiones de Cisco IOS (provisional)	Versiones principales del software Cisco IOS	Versiones del software Cisco IOS (provisional)
dMLP	Chan-PA PA-4T+ PA-8T	12.0T 12.0S 12.1 12.1T 12.2 12.2T 12.3 12.3T 12.2S 12.1E ²	12.0(3)T y posterior 12.0(9) S y posteriores	12.2SX 12.1E ²	—
dLFloLL	Chan-PA PA-4T+ PA-8T	12.2T 12.3 12.3T 12.0S	12.2(8)T y posterior 12.0(24) S y posteriores	12,2SX	12.2(17))SXB y posteriores
dLFloFR	Chan-PA PA-4T+	12.2T 12.3 12.3T	12.2(4)T 3 y	12,2SX	12.2(17))SXB y

	PA-8T		posteriores		posteriores
dLFlo ATM	PA-A3 PA-A6	12.2T 12.3 12.3T	12.2(4)T 3 y posteriores	12,2SX	12.2(17))SXB y posteriores
dMLFR	Chan-PA PA-4T+ PA-8T	12.0S 12.3T	12.0(24) S y posteriores 12.3(4)T y posteriores	12,2SX	12.2(17))SXB y posteriores
QoS en dMLP	Chan-PA PA-4T+ PA-8T	12.0S 12.2T 12.3 12.3T	12.0(24) S y posteriores 12.2(8)T y posteriores	12,2SX	12.2(17))SXB y posteriores
MPLS en dMLP MPLS en dLFloLL	Chan-PA PA-4T+ PA-8T	12.2T 12.3	12.2(15) T11 y posteriores 12.3(5a) y posteriores	12,2SX	12.2(17))SXB y posteriores
DDR distribuido	PA-MC- xT1 PA- MC-xE1 PA-MC- xTE1 PA-MCX- xTE1	12.3T	12.3(7)T y posteriores	—	—

Nota: Tenga en cuenta esta información:

1. Estos PA admiten funciones distribuidas:CT3IPPA-MC-T3PA-MC-2T3+PA-MC-E3PA-MC-2E1PA-MC-2T1PA-MC-4T1PA-MC-8T1PA-MC-8E1PA-MC-8TE1+PA-MC-STM-1
2. La versión 12.1E del software del IOS de Cisco soporta estas funciones en las plataformas 7500 y 7600.

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](#).

Funciones distribuidas

Estas características se explican en este documento:

- MLP distribuido
- LFI distribuido
- LFI distribuido sobre línea arrendada
- LFI distribuido sobre Frame Relay
- LFI distribuido sobre ATM
- Diferencias entre dMLP y dLFioLL
- MLFR distribuido
- Marcador distribuido
- Plataformas y tarjetas de línea que admiten funciones distribuidas

MLPPP distribuido

La función Distributed Multilink Point to Point Protocol (dMLP) permite combinar líneas T1/E1 completas o fraccionarias en una tarjeta de línea (VIP, FlexWAN) en un Cisco 7500 o 7600 Series Router en un paquete que tiene el ancho de banda combinado de varios enlaces. Utilice un paquete MLP distribuido para hacer esto. Un usuario puede elegir el número de paquetes en un router y el número de links por paquete. Esto permite al usuario aumentar el ancho de banda de los links de red más allá del de una sola línea T1/E1 sin necesidad de comprar una línea T3. En non-dMLP, todos los paquetes son conmutados por el Procesador de ruta (RP); por lo tanto, esta implementación impacta el rendimiento del RP, con una alta utilización de la CPU sólo para unas pocas líneas T1/E1 que ejecutan MLP. Con dMLP, aumenta el número total de paquetes que se pueden manejar en el router, ya que la ruta de datos se maneja y se limita mediante la CPU y la memoria de la tarjeta de línea. dMLP le permite empaquetar T1/E1 fraccional, a partir de DS0 (64 Kbps) en adelante.

LFI distribuido

La función dLFI admite el transporte de tráfico en tiempo real (como voz) y tráfico en tiempo no real (como datos) en circuitos virtuales ATM y Frame Relay de menor velocidad (VC) y en líneas arrendadas sin causar un retraso excesivo en el tráfico en tiempo real.

Esta función se implementa mediante PPP de enlaces múltiples (MLP) sobre Frame Relay, ATM y líneas arrendadas. La función fragmenta un paquete de datos grande en una secuencia de fragmentos más pequeños, para permitir que los paquetes en tiempo real sensibles al retardo y los paquetes en tiempo no real compartan el mismo link. Los fragmentos se entrelazan con los paquetes en tiempo real. En el lado receptor del link, los fragmentos se vuelven a montar y el paquete se reconstruye.

La función dLFI es a menudo útil en redes que envían tráfico en tiempo real a través de la cola distribuida de baja latencia (como voz) pero tienen problemas de ancho de banda. Esto retrasa el tráfico en tiempo real debido al transporte de paquetes de datos grandes y menos urgentes. Puede utilizar la función dLFI en estas redes, para desmontar los paquetes de datos grandes en varios segmentos. Los paquetes de tráfico en tiempo real pueden entonces enviarse entre estos segmentos de los paquetes de datos. En este escenario, el tráfico en tiempo real no experimenta

un retraso prolongado mientras espera que los paquetes de datos de baja prioridad atraviesen la red. Los paquetes de datos se reensamblan en el lado receptor del link, por lo que los datos se entregan intactos.

El tamaño del fragmento del link se calcula en base al retraso del fragmento en el agrupamiento de links múltiples, configurado con el comando **ppp multilink fragment-delay *n***, *donde*:

```
fragment size = bandwidth × fragment-delay / 8
```

Este tamaño de fragmento representa solamente la carga útil IP. No incluye los bytes de encapsulación (tamaño del fragmento = peso - bytes de encapsulación). Los términos "peso" y "tamaño del fragmento" se ven en la salida del comando **show ppp multilink** en el RP. Si no se configura el retraso del fragmento, se calcula el tamaño predeterminado del fragmento para un retraso máximo de fragmento de 30.

Nota: Con links de ancho de banda variable, el tamaño del fragmento elegido se basa en el link con el menor ancho de banda.

[LFI distribuido sobre línea arrendada](#)

La función dLFloLL extiende la fragmentación de link distribuido y la funcionalidad de entrelazado a las líneas arrendadas. El LFI distribuido se configura con el comando **ppp multilink interleave** en la interfaz de grupo multilink. Se recomienda que utilice el LFI distribuido en interfaces multilink con un ancho de banda inferior a 768 Kbps. Esto se debe a que la demora de serialización de los paquetes de 1500 bytes para un ancho de banda mayor a 768 Kbps se encuentra dentro de límites de demora aceptables y no necesita fragmentarse.

[LFI distribuido sobre Frame Relay](#)

La función dLFloFR es una extensión de la función PPP de enlaces múltiples sobre Frame Relay (MLPoFR). MLP se utiliza para la fragmentación. Esta función es similar a FRF.12, que admite fragmentación y puede intercalar paquetes de alta prioridad a través de la cola de baja latencia.

El comando **ppp multilink interleave** en la plantilla virtual es necesario para habilitar el entrelazado en la interfaz de acceso virtual asociada. Además de habilitar la conmutación CEF distribuida en la interfaz serial, este comando es un prerequisite para que el LFI distribuido funcione.

Nota: A menos que esté utilizando Frame Relay para la interconexión de redes ATM, se recomienda que utilice FRF.12 en lugar de dLFloFR, porque la utilización del ancho de banda es mejor con FRF.12

[LFI distribuido sobre ATM](#)

La función dLFloATM es una extensión de la función PPP de links múltiples sobre ATM (MLPoATM). MLP se utiliza para la fragmentación.

El comando **ppp multilink interleave** en la plantilla virtual es necesario para habilitar el entrelazado en la interfaz de acceso virtual asociada. Además de habilitar la conmutación CEF distribuida en la interfaz serial, este comando es un prerequisite para que el LFI distribuido funcione.

Con dLFloATM, es muy importante que seleccione un tamaño de fragmento que haga que los

paquetes encajen en las celdas ATM de manera que no causen relleno innecesario en las celdas ATM. Por ejemplo, si el tamaño del fragmento seleccionado es 124 bytes, esto significa que una carga útil IP de 124 bytes finalmente sería $124 + 10$ (encabezado MLP) + 8 (encabezado SNAP) = 142 bytes. Es importante tener en cuenta que el primer fragmento saldría con $124 + 10 + 2$ (tamaño del encabezado PID del primer fragmento) + 8 = 144 bytes. Esto significa que este paquete usará tres celdas ATM para transferir la carga útil y, por lo tanto, usaría la celda empaquetada de la manera más eficiente.

Diferencias entre dMLP y dLFIoLL

dMLP no soporta la fragmentación en el lado de transmisión, mientras que dLFIoLL sí.

Nota: El entrelazado y la fragmentación utilizados con más de un link en el agrupamiento de links múltiples para el tráfico de voz no garantiza que el tráfico de voz recibido a través de links múltiples en el agrupamiento se reciba en orden. El orden correcto de voz se maneja en las capas superiores.

MLFR distribuido

La función MLFR distribuida introduce la funcionalidad basada en el Acuerdo de implementación de Frame Relay Multilink Frame Relay UNI/NNI (FRF.16) del Foro de Frame Relay a los Cisco 7500 y 7600 Series Routers habilitados para tarjeta de línea. La función MLFR distribuida proporciona una manera rentable de aumentar el ancho de banda para aplicaciones concretas porque permite que varios links seriales se agreguen en un solo paquete de ancho de banda. MLFR es compatible con interfaces de usuario a red (UNI) e interfaces de red a red (NNI) en redes Frame Relay.

El paquete está formado por varios links seriales, llamados links de agrupamiento. Cada link de agrupamiento dentro de un agrupamiento corresponde a una interfaz física. Los links de agrupamiento son invisibles para la capa de link de datos de Frame Relay, por lo que la funcionalidad de Frame Relay no se puede configurar en estas interfaces. La funcionalidad regular de Frame Relay que desea aplicar a estos links se debe configurar en la interfaz de agrupamiento. Los enlaces de paquete son visibles para los dispositivos de peer.

DDR distribuido

La función DDR distribuida permite el switching distribuido en las interfaces del marcador. Sin esta función, todo el tráfico de marcado debe enviarse al host para el switching. Con él, sólo los paquetes de control se envían al RP, mientras que la decisión de conmutación se toma en los mismos VIP después de que se ha establecido una conexión.

Tanto la configuración del marcador heredado como la configuración del perfil del marcador se soportan solamente con la encapsulación PPP. MLP en las interfaces del marcador también se soporta. QoS no se soporta con conmutación distribuida en las interfaces del marcador.

Restricciones y requisitos previos de las funciones distribuidas

Prerequisites

Estos son requisitos previos generales para todas estas funciones distribuidas:

- El switching distribuido de Cisco Express Forwarding (dCEF) debe habilitarse globalmente.
- La conmutación dCEF se debe habilitar en la interfaz serial del miembro, que forman parte del conjunto MLP.
- La conmutación dCEF se debe habilitar en el link físico de las interfaces dLFloFR y dLFloATM.
- La configuración de intercalación es necesaria para distribuir LFloFR y LFloATM.
- Configure el ancho de banda requerido en la interfaz de plantilla virtual para las interfaces dLFloFR y dLFloATM.
- Cuando se habilitan las depuraciones PPP en el RP, puede observar el `MLP: Reenviado a un mensaje de interfaz incorrecto` en el Procesador de switch de ruta (RSP). Debido a que este mensaje es confuso y no deseado, especialmente si el mensaje es para los paquetes de Cisco Discovery Protocol (CDP), debe configurar **no cdp enable** en los links miembro del paquete.
- Todos los enlaces de miembro del paquete deben tener el `keepalive` habilitado.

Restricciones

Estas son restricciones generales para todas estas funciones distribuidas:

- Las líneas T1 y E1 no se pueden mezclar en un paquete.
- El retardo diferencial máximo soportado es de 30 ms.
- Todas las líneas de un paquete deben residir en el mismo adaptador de puerto (PA).
- No se admite la compresión de hardware.
- VIP o FlexWAN CEF se limita a IP solamente; todos los demás protocolos se envían al RSP.
- La fragmentación no se soporta en el lado de transmisión para dMLP y dMLFR.
- Muchos de los mecanismos de colocación en cola más antiguos no son soportados por dLFI. Estos mecanismos incluyen: Cola justa en una interfaz de plantilla virtual, Detección aleatoria en una interfaz de plantilla virtual, Cola personalizada, Cola prioritaria.
- Las colas justas, la detección aleatoria (dWRED) y las colas de prioridad se pueden configurar en una política de tráfico con la CLI de QoS modular.
- Sólo se admite un link por paquete MLP, cuando se utiliza dLFloFR o dLFloATM. Si se utiliza más de un link en un paquete MLP cuando se usa dLFloFR o dLFloATM, dLFI se inhabilita automáticamente. Cuando se utiliza dLFI en líneas arrendadas, se puede configurar más de un link con dLFI en el paquete MLP.
- Con dLFloATM, sólo se soportan `aal5snap` y `aal5mux`. Encapsulation `aal5nlpid` y `aal5ciscopp` no son compatibles.
- Sólo se admite voz sobre IP; La función dLFI no admite Voz sobre Frame Relay y Voz sobre ATM.
- La configuración de protocolo comprimido en tiempo real (CRTP) no se debe configurar en la interfaz de enlaces múltiples cuando se utiliza esta combinación de funciones: Interfaz multilink con LFI habilitado, El paquete multilink tiene más de un link miembro, La política de QoS con la función de prioridad se habilita en la interfaz multilink.

Con la configuración dMLP y dLFI, los paquetes de prioridad no llevan el encabezado MLP ni el número de secuencia, y MLP distribuirá los paquetes de prioridad a través de todos los links miembro. Como resultado, los paquetes que son comprimidos por CRTP pueden llegar fuera de servicio al router receptor. Esto prohíbe que CRTP descomprima el encabezado del paquete y fuerza a CRTP a descartar los paquetes.

Recomendaciones

Se recomienda que los links de miembro en un paquete tengan el mismo ancho de banda. Si agrega links de ancho de banda desiguales al paquete, esto llevará a un gran reordenamiento de paquetes, lo que hará que el rendimiento total del paquete disminuya.

Se recomienda utilizar VIP2-50 (con 8 MB de SRAM) o superior con estas funciones distribuidas.

Número de paquetes, enlaces y requisitos de memoria

Consulte [Protocolo Punto a Punto Multilink Distribuido para Cisco 7500 Series Routers](#).

Hardware y software MLPPP o MLFR en tarjetas de línea SIP 7600

MLP y MLFR pueden estar basados en software o hardware. En MLP o MLFR basados en hardware, Freedm proporciona la funcionalidad de links múltiples y los encabezados MLP son agregados por Freedm Chip. En MLP o MLFR basados en software, la CPU de la tarjeta de línea SIP proporciona la funcionalidad multilink (que es similar a las implementaciones de VIP y FlexWAN).

Estas son las limitaciones y condiciones para ejecutar MLP o MLFR basados en hardware.

- Solo puede haber un máximo de 336 paquetes por tarjeta de línea y 168 paquetes por evaluación de estado de seguridad (SPA) (Freedm).
- Sólo puede haber un máximo de 12 DS1/E1 por paquete.
- Todos los enlaces deben pertenecer al mismo SPA (Freedm).
- Todos los links en el paquete deben funcionar a la misma velocidad.
- El tamaño del fragmento TX puede ser 128, 256 o 512. Un tamaño de fragmento configurado de CLI se asigna al tamaño de fragmento admitido más cercano.

```
IF (0 < cli_fragment_size - 6 < 256)
  configured_fragment_size = 128
Else IF (cli_fragment_size < 512)
  configured_fragment_size = 256
Else
  configured_fragment_size = 512
```

- El tamaño del fragmento de RX puede ser de 1 a 9,6 KB.
- No se puede admitir el formato propietario de Cisco (MLFR).

En el LFI de hardware, si sólo hay un link en el paquete y si eso es DS1/E1, la fragmentación y el entrelazado se harán por Freedm.

El resultado de **show ppp multilink** muestra si la implementación del hardware se está ejecutando.

```
Multilink1, bundle name is M1
Bundle up for 00:14:51
Bundle is Distributed

0 lost fragments, 0 reordered, 0 unassigned
0 discarded, 0 lost received, 1/255 load
Member links: 1 active, 0 inactive (max not set, min not set)
Se6/1/0/1:0, since 00:14:51, no frags rcvd
Distributed fragmentation on. Fragment size 512. Multilink in Hardware.
```


Si multilink está basado en software, la salida **show ppp multilink** no tendrá `Multilink en Hardware` en la salida.

Vida de un paquete

Ruta De Datos Rx

1. Paquete recibido por el controlador.
2. La encapsulación está marcada: como sigue:Encapsulación básica:En dMLP, el tipo de encapsulación para la interfaz de ingreso es ET_PPP.En dMLFR, el tipo de encapsulación para la interfaz de ingreso es ET_FRAME_RELAY.En dLFloLL, el tipo de encapsulación para la interfaz de ingreso es ET_PPP.En dLFloFR, el tipo de encapsulación para la interfaz de ingreso es ET_FRAME_RELAY.En dLFloATM, el tipo de encapsulación para la interfaz de ingreso es ET_ATM.En dDialer, el tipo de encapsulación es ET_PPP.Procesamiento de encapsulación adicional:Para ET_PPP, el NLPID se ha rastreado.Para dMLP, el NLPID es MULTILINK.Para dLFloLL, hay dos cosas que se deben tener en cuenta:Paquetes VoIP: no tienen un encabezado MLP y tienen un NLPID que indica IP.Paquetes de datos: el NLPID es MULTILINK.Para dDialer, los paquetes no tendrán un encabezado MLP y tendrán un NLPID que indique IP.**Nota:** En este caso, puede configurar dCRTP (Distributed Compress Real-Time Protocol). Si es así, el encabezado se descomprime antes de procesarlo.
3. Para ET_FRAME_RELAY, si el link en el que se recibe el paquete está configurado para dMLFR, el paquete se procesa para dMLFR
4. Para dLFloFR y dLFloATM, el tipo de encapsulación es ET_FRAME_RELAY y ET_ATM, respectivamente; pero dentro de eso hay un encabezado PPP. El encabezado PPP, como con dLFloLL, indicará si el paquete es un paquete de voz o un paquete de datos. Si se configura dCRTP, el encabezado se descomprime antes de un procesamiento adicional. Los paquetes de voz se conmutan inmediatamente.Un paquete de datos fragmentado tendrá que ser reensamblado antes de que se conmute.Con ET_PPP, puede encontrarse con los paquetes de link PPP; y con ET_FRAME_RELAY, puede encontrarse con los paquetes de control MLFR. Todos estos paquetes de control se envían al RP para su procesamiento.
5. Según la decodificación mencionada, el paquete se verifica para el tipo de conmutación que requiere. El tipo de link determinará si el paquete debe ser conmutado por IP o conmutado por MPLS. Los paquetes se entregan luego a las respectivas funciones de conmutación.
6. Con el agrupamiento en conjunto con las funciones distribuidas, se roba el vector de IP turbo fast switching. Esto se hace porque el paquete se recibe en el link de miembro; sin embargo, debe tratarse de manera que se reciba en el paquete.También debe verificar los paquetes de control que se envían al host. Principalmente en dMLFR, hay paquetes de interfaz de administración local (LMI) que *no* son paquetes de control MLFR. Para estos, se utiliza una parte diferente del espacio de números dLCI. Siempre que el dLCI se descodifica para caer en este espacio, el paquete se envía al host, porque se reconoce que es un paquete LMI.Los paquetes VoIP (en cola en cola de baja latencia) se acaban de apagar sin la adición del encabezado MLP. Las funciones distribuidas pueden recibir y reensamblar paquetes cuando se reciben paquetes de datos fragmentados. El proceso de reensamblado se explica en una sección posterior.Si el paquete se debe conmutar por etiqueta, se pasa a la rutina de switching de etiquetas, en dMLP. De lo contrario, si se va a conmutar por IP, se pasa a la rutina de IP-switching.**Nota:** Todos los paquetes que no son de IP son impulsados al host, en dMLFR.

7. **IP:** La función de IP-switching es común a todos los paquetes. Hace principalmente tres cosas: Realice el procesamiento necesario de los paquetes, en caso de que se configure alguna función. Además, cuando se utiliza Distributed Dialer, haga actualizaciones del temporizador de inactividad aquí cuando se reciba un "paquete interesante". Consulte [dialer idle-timeout \(interface\)](#) , [dialer fast-idle \(interface\)](#) , y [Configuración de un Perfil de Marcador](#) para obtener detalles del parámetro de configuración del [temporizador inactivo](#). En los routers 75xx, la adyacencia indicará el `tx_acc_ptr` para la interfaz de egreso. Si la interfaz de salida es una interfaz de acceso virtual, `tx_acc_ptr` es `NULL`. En este caso, arregle la encapsulación y obtenga el `tx_acc_ptr` del modificador `fib`. Esta búsqueda y corrección de encapsulación es necesaria en `dLFloFR` y `dLFloATM`. En `dLFloLL`, el link se trata como parte de un agrupamiento de links múltiples. **Nota:** TTL para el paquete se ajusta aquí y se realiza la verificación de la fragmentación IP. El `mci_status` se establece en `RXTYPE_DODIP` para todos los paquetes.
8. Con la decisión de switching tomada, el paquete está listo para ser enviado desde la interfaz. La interfaz se verifica para determinar si admite conmutación local. Si lo hace, se envía directamente a través de `fastsend`. De lo contrario, se intenta rutear-cache switch it. Tenga en cuenta que, en caso de que QoS se configure para la interfaz, QoS roba el vector de conmutación local. HQF pondrá en cola el paquete y procesará el paquete más adelante, antes de que finalmente se envíe fuera de la interfaz. Este es el caso con `dLFI`. Para `dLFI`, se establece la fragmentación y el entrelazado. QoS gestiona la invocación de nuestra rutina de fragmentación e intercala los paquetes fragmentados con los paquetes de voz que se pondrán en cola en la cola de prioridad (si `LLQ` está configurado). Esto garantiza que los paquetes VoIP no sufran el retraso necesario para enviar paquetes de datos enormes a través del link.

[Ruta de datos Tx](#)

`vip_dtq_Consumer` obtiene el paquete y obtiene el número de interfaz, desde el cual obtiene el `idb`. La rutina `fastsend` que corresponde al `idb` se llama:

i) Fastmanda

1. En `dMFR`, la estructura `fr_info` se recupera de la tabla que coincide con `if_index` en `fr_info`. Los paquetes de control se acaban de enviar. El encabezado de trama le dará el `dLCI`, lo que le ayudará a determinar si se trata de un paquete LMI o de un paquete de datos. El campo `dlci` del encabezado de trama se sobrescribe con el número de secuencia `dmfr`. Los números de secuencia separados se utilizan para LMI y paquetes de datos. **Nota:** Se utilizan números de secuencia separados para `dLCI` separados.
2. En `dMLP`, los paquetes de control se envían con la prioridad establecida en alta. Con los paquetes de datos, si se configura `dCRTP`, el encabezado se comprime. El encabezado MLP VIP que incluye la información de secuenciación se agrega y se envía desde los enlaces de miembro.
3. En `dLFI`, HQF intercepta los paquetes que se enviarán a través de la interfaz. Si se trata de un paquete de voz, el paquete de voz se coloca en la cola de prioridad (si se configura `LLQ`) y se envía fuera de la interfaz sin la encapsulación MLP. Con los paquetes de datos, llama al código de fragmentación `dLFI`, que devuelve fragmentos al código QoS, que luego se entrelazan con el tráfico de prioridad para que se cumplan los requisitos de demora del tráfico de voz. Además, si se configura `dCRTP`, sólo se comprime el encabezado del

paquete de voz. Los encabezados de los paquetes de datos se dejan tal cual.

4. En dDialer, el paquete se clasifica para restablecer el temporizador de inactividad del link de salida antes de que se envíe el paquete. Esto se hace después de elegir el link de salida, en el caso de que varios links estén enlazados al mismo marcador. No se agrega ningún encabezado a los paquetes del marcador. Por lo tanto, la secuenciación y reensamblado de paquetes no se soportan en las interfaces del marcador.

Nota: En dMLP, dDialer, dMLFR y dLFI con varios links, el link físico en el que se reenvía el tráfico depende de la congestión del link. Si el link está congestionado, pase al siguiente link y así sucesivamente. (dMLFR, dMLP sin QoS, y las funciones dDialer también eligen links según el número de bytes que se ponen en el link. Selecciona el siguiente link, si el link actual ya ha transmitido su cuota de bytes, en base a ordenamiento cíclico. Esta cuota se decide por frag_bytes para el link. Para las interfaces miembro del marcador, frag_bytes se establece en el valor predeterminado del ancho de banda de la interfaz.)

Nota: En las configuraciones HQF en las interfaces del VIP de salida, HQF roba el vector dtq_Consumer. El DMA de paquete al VIP de salida primero pasa por la verificación HQF. Si QoS se configura en la interfaz de salida, HQF inicia para procesar el paquete, antes de que el paquete se envíe rápidamente fuera de la interfaz.

Reensamblado

Las interfaces de marcador sencillo no admiten el reensamblado ni la secuenciación. Para habilitar esto en las interfaces del marcador, se deberá configurar MLP sobre las interfaces del marcador. Si esto se hace, la trayectoria Rx y Tx son idénticas a las trayectorias dMLP. Cuando se reciben los paquetes, se verifica el número de secuencia con el número de secuencia esperado.

- Si los números de secuencia coinciden: Si el paquete es un paquete no fragmentado, no se requiere el reensamblado. Continúe con los pasos de conmutación adicionales. Si el paquete es un fragmento, verifique los bits inicial y final y construya el paquete como y cuando se reciban los fragmentos.
- Si los números de secuencia no coinciden: Si el número de secuencia está dentro de la ventana esperada de números de secuencia, colóquelo en la "lista de fragmentos sin asignar" ordenada. Más tarde, cuando no se recibe un número de secuencia esperado, se verifica esta lista, en caso de que el paquete se almacenara aquí. Si el número de secuencia no está dentro de la ventana, deséchelo e informe "fragmento perdido recibido". Si un tiempo de espera se produce más tarde mientras se espera este paquete, el receptor se resincroniza y comienza de nuevo con el siguiente paquete recibido.

En todos esos casos, se envía un flujo de paquetes ordenado correctamente desde esta interfaz. Si se reciben fragmentos, se forma un paquete completo y luego se envía.

Configuración, Verificación y Depuración de Funciones Distribuidas

Esta sección cubre los comandos **show** y **debug** que están disponibles para verificar y depurar cada una de las funciones distribuidas.

Configuración y Verificación de dMFR

Ejemplo de configuración MFR

```
interface MFR1
  no ip address

interface MFR1.1 point-to-point
  ip address 181.0.0.2 255.255.0.0
  frame-relay interface-dlci 16
```

Nota: La interfaz MFR es como otra interfaz FR y, por lo tanto, admite la mayor parte de la configuración FR.

```
interface Serial5/0/0/1:0
  no ip address
  encapsulation frame-relay MFR1
  tx-queue-limit 26

interface Serial5/0/0/2:0
  no ip address
  encapsulation frame-relay MFR1
  tx-queue-limit 26

interface Serial5/0/0/3:0
  no ip address
  encapsulation frame-relay MFR1
```

Verifique el estado del paquete MFR en el RP

show frame-relay multilink

```
Bundle: MFR1, State = up, class = A, fragmentation disabled
BID = MFR1
Bundle links:
  Serial5/0/0/3:0, HW state = up, link state = Add_sent, LID = Serial5/0/0/3:0
  Serial5/0/0/2:0, HW state = up, link state = Up, LID = Serial5/0/0/2:0
  Serial5/0/0/1:0, HW state = up, link state = Up, LID = Serial5/0/0/1:0
```

Esto indica que dos interfaces se agregan correctamente y una interfaz aún no ha negociado los mensajes LIP MLFR.

Para obtener más información sobre el paquete MFR y los links de miembro, ejecute este comando:

show frame-relay multilink mfr1 detailed

```
Bundle: MFR1, State = up, class = A, fragmentation disabled
BID = MFR1
No. of bundle links = 3, Peer's bundle-id = MFR1
Rx buffer size = 36144, Lost frag timeout = 1000
Bundle links:
  Serial5/0/0/3:0, HW state = up, link state = Add_sent, LID = Serial5/0/0/3:0
    Cause code = none, Ack timer = 4, Hello timer = 10,
    Max retry count = 2, Current count = 0,
    Peer LID = , RTT = 0 ms
  Statistics:
    Add_link sent = 35, Add_link rcv'd = 0,
```

```
Add_link ack sent = 0, Add_link ack rcv'd = 0,
Add_link rej sent = 0, Add_link rej rcv'd = 0,
Remove_link sent = 0, Remove_link rcv'd = 0,
Remove_link_ack sent = 0, Remove_link_ack rcv'd = 0,
Hello sent = 0, Hello rcv'd = 0,
Hello_ack sent = 0, Hello_ack rcv'd = 0,
outgoing pak dropped = 0, incoming pak dropped = 0
Serial5/0/0/2:0, HW state = up, link state = Up, LID = Serial5/0/0/2:0
Cause code = none, Ack timer = 4, Hello timer = 10,
Max retry count = 2, Current count = 0,
Peer LID = Serial6/1/0/2:0, RTT = 32 ms
Statistics:
Add_link sent = 0, Add_link rcv'd = 0,
Add_link ack sent = 0, Add_link ack rcv'd = 0,
Add_link rej sent = 0, Add_link rej rcv'd = 0,
Remove_link sent = 0, Remove_link rcv'd = 0,
Remove_link_ack sent = 0, Remove_link_ack rcv'd = 0,
Hello sent = 7851, Hello rcv'd = 7856,
Hello_ack sent = 7856, Hello_ack rcv'd = 7851,
outgoing pak dropped = 0, incoming pak dropped = 0
Serial5/0/0/1:0, HW state = up, link state = Up, LID = Serial5/0/0/1:0
Cause code = none, Ack timer = 4, Hello timer = 10,
Max retry count = 2, Current count = 0,
Peer LID = Serial6/1/0/1:0, RTT = 32 ms
Statistics:
Add_link sent = 0, Add_link rcv'd = 0,
Add_link ack sent = 0, Add_link ack rcv'd = 0,
Add_link rej sent = 0, Add_link rej rcv'd = 0,
Remove_link sent = 0, Remove_link rcv'd = 0,
Remove_link_ack sent = 0, Remove_link_ack rcv'd = 0,
Hello sent = 7851, Hello rcv'd = 7856,
Hello_ack sent = 7856, Hello_ack rcv'd = 7851,
outgoing pak dropped = 0, incoming pak dropped = 0
```

Comandos de depuración MFR

Estas depuraciones son útiles para resolver problemas en los que no se agrega un link al paquete.

```
debug frame-relay multilink control
```

Nota: Cuando no se especifica una interfaz MFR específica o una interfaz serial, esto habilita la depuración para todos los links MFR. Esto puede ser abrumador, si el router tiene un gran número de links MFR.

Para depurar los paquetes MFR recibidos en el RP, así como para depurar las actividades de control MFR, esta depuración es útil:

```
debug frame-relay multilink
```

Nota: En el tráfico denso, esto puede saturar la CPU.

Verifique el estado del paquete dMLFR en la LC

```
show frame-relay multilink
```

Nota: Actualmente, esto no está disponible en la LC, pero pronto será agregado. Hasta entonces, use **show ppp multilink**.

```
Bundle MFR1, 2 members
  bundle 0x62DBDD20, frag_mode 0
  tag vectors 0x604E8004 0x604C3628
  Bundle hwidb vector 0x6019271C
  idb MFR1, vc 0, RSP vc 0
  QoS disabled, fastsend (mlp_fastsend), visible_bandwidth 3072
  board_encap 0x60577554, hw_if_index 0, pak_to_host 0x0
  max_particles 400, mrru 1524, seq_window_size 0x200
  working_pak 0x0, working_pak_cache 0x0
  una_frag_list 0x0, una_frag_end 0x0, null_link 0
  rcved_end_bit 1, is_lost_frag 0, resync_count 0
  timeout 0, timer_start 0, timer_running 0, timer_count 0
  next_xmit_link Serial0/0:1, member 0x3, congestion 0x3
dmlp_orig_pak_to_host 0x603E7030
dmlp_orig_fastsend 0x6035DBC0
bundle_idb->lc_ip_turbo_fs 0x604A7750
  0 lost fragments, 0 reordered, 0 unassigned
  0 discarded, 0 lost received
  0x0 received sequence, 0x58E sent sequence
DLCI: 16
  769719 lost fragments, 22338227 reordered,
                        0 unassigned
  27664 discarded, 27664 lost received
  0xF58 received sequence, 0x8DE sent sequence
timer count 767176
  Member Link: 2 active
    Serial0/0:0, id 0x1, fastsend 0x60191E34, lc_turbo 0x601913AC, PTH 0x603E7030, OOF 0
    Serial0/0:1, id 0x2, fastsend 0x60191E34, lc_turbo 0x601913AC, PTH 0x603E7030, OOF 0
```

[Configuración y Verificación de dMLP/dLFloLL](#)

[Configuración PPP Multilink](#)

```
interface Multilink1
  ip address 151.0.0.2 255.255.0.0
  no cdp enable
  ppp chap hostname M1
  ppp multilink
```

!

Ejemplo de configuración en la interfaz serial:

```
interface Serial5/0/0/4:0
  no ip address
  encapsulation ppp
  tx-queue-limit 26
  no cdp enable
  ppp chap hostname M1
  ppp multilink group 1
```

!

```
interface Serial5/0/0/5:0
  no ip address
  encapsulation ppp
  tx-queue-limit 26
  no cdp enable
  ppp chap hostname M1
```

```
ppp multilink group 1
```

```
!
```

Nota: El comando **ppp chap hostname M1** no significa realmente que la autenticación CHAP esté habilitada. La cadena **M1** en este comando actúa como el discriminador de punto final y sólo es necesaria si va a haber más de un agrupamiento de links múltiples entre los mismos dos routers. En tal caso, todos los links que pertenecen a un paquete deben tener el mismo discriminador de punto final, y ninguno de los dos links que pertenecen a un paquete diferente debe tener el mismo discriminador de punto final.

Parámetros de configuración opcionales

[no] ppp multilink interleave

Esto habilita el entrelazado en el agrupamiento de links múltiples. Esto funciona junto con la CLI de QoS modular. Los paquetes de alta prioridad se transmitirán sin la adición de la secuencia y el encabezado MLP, mientras que otros paquetes se fragmentarán y transmitirán con la secuencia y el encabezado MLP.

Nota: Cuando el entrelazado está habilitado con más de un link, es posible que el tráfico de alta prioridad se vuelva a ordenar. Cuando el entrelazado está activado o desactivado, se requiere un reinicio del paquete para activarlo en la tarjeta de línea.

```
ppp multilink mrru local value
```

Esto especifica la unidad máxima de recepción en el link múltiple; los paquetes de hasta este tamaño serán aceptados por la interfaz multilink. El tamaño aquí excluye el encabezado MLP.

```
ppp multilink mrru remote value
```

Esto especifica la MRU mínima que el extremo remoto debe soportar. Si la MRU del extremo remoto es menor que este valor, la negociación del agrupamiento fallará.

```
ppp multilink fragment delay seconds
```

Esto especifica el retraso permitido en milisegundos (ms) causado por un fragmento de datos. En otras palabras, el valor del retraso se utiliza para calcular el tamaño máximo del fragmento permitido. La implementación distribuida difiere de la implementación de Cisco IOS de las siguientes maneras:

1. La fragmentación no se realiza a menos que esté habilitado el entrelazado.
2. Con links de ancho de banda variable, el tamaño del fragmento elegido se basa en la interfaz de menor ancho de banda.

```
ppp multilink fragment disable
```

Este comando no agrega ninguna funcionalidad en la implementación distribuida. La fragmentación sólo se produce cuando se habilita el entrelazado; y, cuando se habilita el entrelazado, el comando **ppp multilink fragment disable** se ignora.

[Verifique el estado del paquete dMLP en el RP](#)

```
show ppp multilink
```

```
Multilink1, bundle name is M1
Endpoint discriminator is M1
Bundle up for 00:09:09, 1/255 load
Receive buffer limit 24000 bytes, frag timeout 1000 ms
Bundle is Distributed
  0/0 fragments/bytes in reassembly list
  0 lost fragments, 0 reordered
  0/0 discarded fragments/bytes, 0 lost received
  0x9 received sequence, 0x0 sent sequence
dLFI statistics:
      DLFI Packets   Pkts In   Chars In   Pkts Out   Chars Out
      Fragmented    0         0         0         0
      UnFragmented  9        3150      0         0
      Reassembled   9        3150      0         0
      Reassembly Drops 0
      Fragmentation Drops 0
      Out of Seq Frags 0
Member links: 2 active, 0 inactive (max not set, min not set)
Se5/0/0/4:0, since 00:09:09, 768 weight, 760 frag size
Se5/0/0/5:0, since 00:09:09, 768 weight, 760 frag size
```

1. Cuando el paquete está en modo distribuido, esto se muestra en la salida de **show ppp multilink**: El paquete está distribuido. Si no es así, el paquete por alguna razón no se distribuye.
2. Cuando se configura y habilita **ppp multilink interleave** en la tarjeta de línea, la salida **show ppp multilink** incluye las estadísticas dLFI donde: **Fragmentado**: indica el recuento de fragmentos que se transmitieron y recibieron. **Desfragmentado**: indica el recuento de paquetes que se transmitieron o recibieron sin fragmentarse. **Reensamblado**: indica el número de paquetes completos que se reensamblaron. Cuando el entrelazado no está habilitado, el resultado es similar a lo siguiente:

```
Multilink1, bundle name is M1
Endpoint discriminator is M1
Bundle up for 00:00:00, 0/255 load
Receive buffer limit 24000 bytes, frag timeout 1000 ms
Bundle is Distributed
  0/0 fragments/bytes in reassembly list
  0 lost fragments, 0 reordered
  0/0 discarded fragments/bytes, 0 lost received
  0x0 received sequence, 0x2 sent sequence
Member links: 2 active, 0 inactive (max not set, min not set)
Se5/0/0/5:0, since 00:00:00, 768 weight, 760 frag size
Se5/0/0/4:0, since 00:00:03, 768 weight, 760 frag size
```

El tamaño del fragmento en el ejemplo anterior es de 760 bytes.

[Verifique el estado del paquete dMLP en la LC](#)

show ppp multilink

```
dmlp_ipc_config_count 24
dmlp_bundle_count 2
dmlp_ipc_fault_count 1
dmlp_il_inst 0x60EE4340, item count 0
0, store 0, hwidb 0x615960E0, bundle 0x622AA060, 0x60579290, 0x6057A29C
1, store 0, hwidb 0x615985C0, bundle 0x622AA060, 0x60579290, 0x6057A29C
2, store 0, hwidb 0x0, bundle 0x0,
3, store 0, hwidb 0x0, bundle 0x0,
4, store 0, hwidb 0x0, bundle 0x0,
5, store 0, hwidb 0x0, bundle 0x0,
6, store 0, hwidb 0x0, bundle 0x0,
7, store 0, hwidb 0x0, bundle 0x0,
8, store 0, hwidb 0x0, bundle 0x0,
9, store 0, hwidb 0x0, bundle 0x0,
```

```
Bundle Multilink1, 2 members
  bundle 0x622AA060, frag_mode 0
  tag vectors 0x604E8004 0x604C3628
  Bundle hwidb vector 0x6057B198
  idb Multilink1, vc 4, RSP vc 4
  QoS disabled, fastsend (qos_fastsend), visible_bandwidth 3072
  board_encap 0x60577554, hw_if_index 0, pak_to_host 0x0
  max_particles 400, mrru 1524, seq_window_size 0x8000
  working_pak 0x0, working_pak_cache 0x0
  una_frag_list 0x0, una_frag_end 0x0, null_link 0
  rcved_end_bit 1, is_lost_frag 1, resync_count 0
  timeout 0, timer_start 0, timer_running 0, timer_count 1
  next_xmit_link Serial0/0:3, member 0x3, congestion 0x3
dmlp_orig_pak_to_host 0x603E7030
dmlp_orig_fastsend 0x6035DBC0
bundle_idb->lc_ip_turbo_fs 0x604A7750
  0 lost fragments, 0 reordered, 0 unassigned
  0 discarded, 0 lost received
  0xC3 received sequence, 0x0 sent sequence
  Member Link: 2 active
    Serial0/0:4, id 0x1, fastsend 0x60579290, lc_turbo 0x6057A29C, PTH 0x60579A18, OOF 0
    Serial0/0:3, id 0x2, fastsend 0x60579290, lc_turbo 0x6057A29C, PTH 0x60579A18, OOF 0
```

Con dMFR, los números de secuencia se mantienen por dLCI, con el número de secuencia en el agrupamiento utilizado para LMI dLCI.

Campo	Descripción
dmlp_ipc_config_count	Número de mensajes IPC recibidos por la LC para la configuración de links múltiples o MLFR
dmlp_bundle_count	Número de paquetes MLP y MLFR en la LC
dmlp_ipc_fail_count	Número de mensajes de configuración que resultaron en falla en LC. Debe ser 0; si no es cero, entonces podría haber un problema.
vectores de etiquetas	Indica los vectores idb a tag_optimum_fs y idb a

	ip2tag_optimum_fs utilizados en el switching de etiquetas.
board_encap	Indica el vector board_encap que se utiliza para agregar 2 bytes de encapsulación de placa, si hay links canalizados en una plataforma 7500. Debe ser NULL, si el link contiene interfaces no canalizadas.
max_partículas	Número máximo de partículas que se pueden mantener en el búfer de reensamblado
mrru	El tamaño máximo de paquete que se acepta sin considerar la encapsulación MLP. No aplicable para la interfaz MLFR.
seq_window_size	El tamaño máximo de la ventana para los números de secuencia
working_pak	Indica el paquete actual en reensamblado. NULL, si no hay ninguno.
working_pak_cache	Puntero al paquete estático que se utiliza para el reensamblado. Esto se asigna cuando el paquete recibe el primer paquete no completo.
una_frag_list	Primera entrada en la cola de reensamblado. Si la entrada no es NULL y no cambia, indica que el temporizador no está ejecutando un problema de software.
una_frag_end	Última entrada en la cola de reensamblado
rcved_end_bit	Indica que el paquete ha recibido un bit final, por lo que busca un bit inicial.
is_loss_frag	Es true si se declara perdido un fragmento. Esto se borra cuando se recibe un fragmento con la secuencia esperada.
resync_count	Indica el número de veces que el receptor no estaba sincronizado con el transmisor y tuvo que sincronizarse comenzando con el último fragmento secuenciado recibido.
tiempo de espera	Indica que se ha agotado el tiempo de espera del reensamblado y que los

	paquetes se están procesando desde la cola de reensamblado.
timer_start	Número de veces que se ha iniciado el temporizador de reensamblado
timer_running	Indica si se está ejecutando o no el temporizador de reensamblado.
contador_temporizador	Indica el número de veces que el temporizador de reensamblado ha caducado.
next_xmit_link	El link en el que se transmitirá el siguiente paquete
Miembro	Campo Bit que indica los miembros presentes.
Congestión	Campo no utilizado en todas las sucursales. Indica qué enlaces de miembro no están congestionados.
dmlp_orig_pak_to_host	El vector utilizado para enviar paquetes al RP.
dmlp_orig_fastsend	El envío rápido del controlador original antes de que MLP o MLFR modificaran el envío rápido del controlador.
fragmentos perdidos	Número de fragmentos perdidos (el receptor no recibió estos fragmentos). Esto se borra periódicamente cuando se envía una actualización al host.
Reordenado	Número de fragmentos que se recibieron por orden esperado. Esto se borra periódicamente cuando se envía una actualización al host.
Descartado	Número de fragmentos descartados porque no se pudo realizar un paquete completo
perdido recibido	Número de fragmentos recibidos que se consideró perdidos. Esto indica que la demora entre links es mayor que el tiempo de espera de reensamblado dMLP de 30 ms.

[Configuración y Verificación de dLFloFR y dLFloATM](#)

```
class-map voip
  match ip precedence 3
```

```

policy-map llq
  class voip
    priority

int virtual-template1
  service-policy output llq
  bandwidth 78
  ppp multilink
  ppp multilink interleave
  ppp multilink fragment-delay 8

int serial5/0/0/6:0
  encapsulation frame-relay
  frame-relay interface-dlci 16 ppp virtual-template1
!--- Or

int ATM4/0/0
  no ip address
int ATM4/0/0.1 point-to-point
  pvc 5/100
  protocol ppp virtual-template 1

```

[Verifique el estado del paquete dLFIoFR/ATM en el RP](#)

show ppp multilink

```

Virtual-Access3, bundle name is dLFI
Endpoint discriminator is dLFI
Bundle up for 00:01:11, 1/255 load
Receive buffer limit 12192 bytes, frag timeout 1524 ms
Bundle is Distributed
  0/0 fragments/bytes in reassembly list
  0 lost fragments, 0 reordered
  0/0 discarded fragments/bytes, 0 lost received
  0x0 received sequence, 0x0 sent sequence
dLFI statistics:
      DLFI Packets   Pkts In   Chars In   Pkts Out   Chars Out
      Fragmented           0           0           0           0
      UnFragmented         0           0           0           0
      Reassembled          0           0           0           0
      Reassembly Drops      0
      Fragmentation Drops   0
      Out of Seq Frags      0
Member links: 1 (max not set, min not set)
Vi2, since 00:01:11, 240 weight, 230 frag size

```

Nota: El paquete se distribuirá solamente cuando **ppp multilink interleave** se configure bajo la plantilla virtual; sin este comando, el paquete no se distribuirá.

[Verifique el estado del paquete dLFIoFR/ATM en la LC](#)

Para verificar que el dLFI esté funcionando correctamente en la LC, ejecute este comando:

show hqf interface

Interface Number 6 (type 22) Serial0/0:5

blt (0x62D622E8, index 0, hwidb->fast_if_number=35) layer PHYSICAL
scheduling policy: FIFO
classification policy: NONE
drop policy: TAIL
blt flags: 0x0

qsize 0 txcount 3 drops 0 qdrops 0 nobuffers 0
aggregate limit 16 individual limit 4 availbuffers 16
weight 1 perc 0.00 ready 1 shape_ready 1 wfq_clitype 0
visible_bw 64 allocated_bw 64 qlimit_tuned 0 vc_encap 2
quantum 1500 credit 0 backpressure_policy 0 nothingoncalQ 1

next layer HQFLAYER_FRAMEDLCI_IFC (max entries 1024)

blt (0x62D620E8, index 0, hwidb->fast_if_number=35) layer FRAMEDLCI_IFC
scheduling policy: FIFO
classification policy: NONE
drop policy: TAIL
blt flags: 0x0

qsize 0 txcount 1 drops 0 qdrops 0 nobuffers 0
aggregate limit 16 individual limit 4 availbuffers 16
weight 1 perc 0.00 ready 1 shape_ready 1 wfq_clitype 0
visible_bw 64 allocated_bw 64 qlimit_tuned 0 vc_encap 2
quantum 1500 credit 0 backpressure_policy 0 nothingoncalQ 1

blt (0x62D621E8, index 16, hwidb->fast_if_number=35) layer FRAMEDLCI_IFC
scheduling policy: WFQ
classification policy: PRIORITY_BASED
drop policy: TAIL
frag policy: root
blt flags: 0x0

qsize 0 txcount 2 drops 0 qdrops 0 nobuffers 0
aggregate limit 16 individual limit 4 availbuffers 16
weight 1 perc 0.00 ready 1 shape_ready 1 wfq_clitype 0
visible_bw 64 allocated_bw 64 qlimit_tuned 0 vc_encap 2
quantum 240 credit 0 backpressure_policy 0 nothingoncalQ 1

next layer HQFLAYER_PRIORITY (max entries 256)

blt (0x62D61FE8, index 0, hwidb->fast_if_number=35) **layer PRIORITY**
scheduling policy: FIFO
classification policy: NONE
drop policy: TAIL
frag policy: leaf
blt flags: 0x0

qsize 0 txcount 0 drops 0 qdrops 0 nobuffers 0
aggregate limit 8 individual limit 2 availbuffers 8
weight 0 perc 0.99 ready 1 shape_ready 1 wfq_clitype 0
visible_bw 32 allocated_bw 32 qlimit_tuned 0 vc_encap 2
quantum 240 credit 0 backpressure_policy 0 nothingoncalQ 1

blt (0x62D61CE8, index 1, hwidb->fast_if_number=35) **layer PRIORITY**
scheduling policy: FIFO
classification policy: NONE
drop policy: TAIL
blt flags: 0x0

Priority Conditioning enabled

qsize 0 txcount 0 drops 0 qdrops 0 nobuffers 0
aggregate limit 0 individual limit 0 availbuffers 0

```
weight 1 perc 0.00 ready 1 shape_ready 1 wfq_clitype 0
visible_bw 0 allocated_bw 0 qlimit_tuned 0 vc_encap 2
quantum 240 credit 0 backpressure_policy 0 nothingoncalQ 1
```

```
PRIORITY: bandwidth 32 (50%)
        last 0 tokens 1500 token_limit 1500
```

```
blt (0x62D61EE8, index 255, hwidb->fast_if_number=35) layer PRIORITY
scheduling policy: WFQ
classification policy: CLASS_BASED
drop policy: TAIL
frag policy: MLPPP (1)
    frag size: 240, vc encaps: 0, handle: 0x612E1320
blt flags: 0x0
```

```
qsize 0 txcount 2 drops 0 qdrops 0 nobuffers 0
aggregate limit 8 individual limit 2 availbuffers 8
weight 1 perc 0.01 ready 1 shape_ready 1 wfq_clitype 0
visible_bw 32 allocated_bw 32 qlimit_tuned 0 vc_encap 2
quantum 1 credit 0 backpressure_policy 0 nothingoncalQ 1
```

```
next layer HQFLAYER_CLASS_HIER0 (max entries 256)
```

```
blt (0x62D61DE8, index 0, hwidb->fast_if_number=35) layer CLASS_HIER0
scheduling policy: FIFO
classification policy: NONE
drop policy: TAIL
frag policy: leaf
blt flags: 0x0
```

```
qsize 0 txcount 2 drops 0 qdrops 0 nobuffers 0
aggregate limit 8 individual limit 2 availbuffers 8
weight 1 perc 50.00 ready 1 shape_ready 1 wfq_clitype 0
visible_bw 32 allocated_bw 32 qlimit_tuned 0 vc_encap 2
quantum 240 credit 0 backpressure_policy 0 nothingoncalQ 1
```

Debe haber una capa de prioridad y una capa WFQ. La fragmentación se realizará en la capa de hoja WFQ blt.

[Configuración y Verificación de dDDR](#)

El DDR distribuido se activa cuando habilita **ip cef distribuido** en la configuración global y **ip route-cache distribuido** en las interfaces del marcador.

```
!--- Global configuration that enables distributed CEF switching. ip cef distributed --- Enable
distributed switching on the dialer interface (on the D-channel interface). int serial 3/1/0:23
ip route-cache distributed !--- Or, enable it on the dialer interface. int Dialer1 ip route-
cache distributed
```

No hay otras configuraciones especiales para DDR distribuido. Una configuración posterior sigue la configuración normal de DDR.

[Verificación del enrutamiento de marcado distribuido a demanda](#)

```
BOX2002# show isdn status
```

```
Global ISDN Switchtype = primary-net5
ISDN Serial3/1/0:23 interface
```

```
--- Network side configuration. dsl 0, interface ISDN Switchtype = primary-net5 Layer 1 Status: ACTIVE Layer 2 Status: TEI = 0, Ces = 1, SAPI = 0, State = MULTIPLE_FRAME_ESTABLISHED
```

The ISDN status should be MULTIPLE_FRAME_ESTABLISHED. This means that the physical layer is ready for ISDN connectivity. Layer 3 Status: 0 Active Layer 3 Call(s) Active dsl 0 CCBs = 0 The Free Channel Mask: 0x807FFFFFF Number of L2 Discards = 0, L2 Session ID = 6 EDGE# **show dialer**

```
Serial6/0:0 - dialer type = ISDN
Idle timer (120 secs), Fast idle timer (20 secs)
Wait for carrier (30 secs), Re-enable (15 secs)
Dialer state is data link layer up
Time until disconnect 119 secs
Current call connected never
Connected to 54321
```

```
Serial6/0:1 - dialer type = ISDN
Idle timer (120 secs), Fast idle timer (20 secs)
Wait for carrier (30 secs), Re-enable (15 secs)
Dialer state is idle
```

El tipo de marcador nos indica el tipo de marcador utilizado. ISDN implica la configuración del marcador heredado y PROFILE implica la configuración del perfil del marcador. El estado del marcador indica el estado actual del marcador. El estado de una interfaz de marcador no conectada estará inactivo. El temporizador de inactividad se reinicia cada vez que se ve tráfico interesante. Si este temporizador alguna vez caduca, la interfaz se desconectará inmediatamente. El temporizador de inactividad es un parámetro configurable. Para obtener más información, consulte [Configuración de DDR de igual a igual con perfiles de marcador](#).

show ppp multilink

```
!--- From LC for dialer profile. dmlp_ipc_config_count 2 dmlp_bundle_count 1 dmlp_il_inst
0x60EE4340, item count 0 0, store 0, hwidb 0x0, bundle 0x0, 1, store 0, hwidb 0x0, bundle 0x0,
2, store 0, hwidb 0x0, bundle 0x0, 3, store 0, hwidb 0x0, bundle 0x0, 4, store 0, hwidb 0x0,
bundle 0x0, 5, store 0, hwidb 0x0, bundle 0x0, 6, store 0, hwidb 0x0, bundle 0x0, 7, store 0,
hwidb 0x0, bundle 0x0, 8, store 0, hwidb 0x0, bundle 0x0, 9, store 0, hwidb 0x0, bundle 0x0,
Bundle Dialer1, 1 member bundle 0x62677220, frag_mode 0 tag vectors 0x604E8004 0x604C3628 Bundle
hwidb vector 0x0 idb Dialer1, vc 22, RSP vc 22 QoS disabled, fastsend (mlp_fastsend),
visible_bandwidth 56 board_encap 0x60577554, hw_if_index 0, pak_to_host 0x0 max_particles 200,
mrru 1524, seq_window_size 0x8000 working_pak 0x0, working_pak_cache 0x0 una_frag_list 0x0,
una_frag_end 0x0, null_link 0 rcved_end_bit 1, is_lost_frag 0, resync_count 0 timeout 0,
timer_start 0, timer_running 0, timer_count 0 next_xmit_link Serial1/0:22, member 0x1,
congestion 0x1 dmlp_orig_pak_to_host 0x603E7030 dmlp_orig_fastsend 0x60381298 bundle_idb-
>lc_ip_turbo_fs 0x604A7750 0 lost fragments, 0 reordered, 0 unassigned 0 discarded, 0 lost
received 0x0 received sequence, 0x0 sent sequence Member Link: 1 active Serial1/0:22, id 0x1,
fastsend 0x60579290, lc_turbo 0x6057A29C, PTH 0x60579A18, OOF 0
```

Las variables que se muestran son las mismas que para dMLP.

[Depuración dMLP y dDDR](#)

[Depuraciones disponibles en el RP](#)

dDDR

```
debug dialer [events | packets | forwarding | map]
```

Ejecute este comando para depurar las funciones de trayectoria de control como la configuración de llamadas, etc. Para obtener más información, refiérase a [eventos debug dialer](#) .

```
debug ip cef dialer
```

Ejecute este comando para depurar eventos de marcador relacionados con CEF. Para obtener más información, consulte [Marcador CEF](#).

[Depuraciones disponibles en la LC](#)

dMPLP

Depuración de Trayectoria de Control: **debug multilink event**

Depuración de ruta de datos: **debug multilink fragments**

Depuración de errores de ruta de datos y ruta de control: **debug multilink error**

Depuración de dMPLP en tarjetas de línea SIP

Dumping de paquetes basados en CI: Los paquetes de datos y los paquetes de control se pueden descartar en tarjetas de línea basadas en la ci de control y la ci de secuencia.

```
test hw-module subslot_num dump ci CI-NUM [rx|tx] num_packets_to_dump
```

Las CI pueden obtenerse de esta manera:

```
!--- Issue show controller serial interface for CTE1.
```

```
SIP-200-6# show controller serial 6/0/0:0
```

```
SPA 6/0 base address 0xB8000000 efc 1
```

```
Interface Serial6/0/0:0 is administratively down
```

```
Type 0xD Map 0x7FFFFFFF, Subrate 0xFF, mapped 0x1, maxmtu 0x5DC
```

```
Mtu 1500, max_buffer_size 1524, max_pak_size 1608 enc 84
```

```
ROM rev: 0, FW OS rev: 0x00000000 Firmware rev: 0x00000000
```

```
idb=0x42663A30, pa=0x427BF6E0, vip_fci_type=0, port_per_spa=0
```

```
SPA port type is set
```

```
Host SPI4 in sync
```

```
SPA=0x427BF6E0 status=00010407, host=00000101, fpga=0x427EDF98
```

```
cmd_head=113, cmd_tail=113, ev_head=184, ev_tail=184
```

```
ev_dropped=0, cmd_dropped=0
```

```
!--- Start Link Record Information. tag 0, id 0, anyphy 0, anyphy_flags 3, state 0
```

```
crc 0, idle 0, subrate 0, invert 0, priority 0
```

```
encap hdlc
```

```
corrupt_ci 65535, transparent_ci 1
```

```
!--- End Link Record Information. Interface Serial6/0/0:0 is administratively down Channel
```

```
Stats: in_throttle=0, throttled=0, unthrottled=0, started=1 rx_packets=0, rx_bytes=0,
```

```
rx_frame_aborts=0, rx_crc_errors=0 rx_giants=0, rx_non_aligned_frames=0, rx_runts=0,
```

```
rx_overruns=0 tx_packets=0, tx_bytes=0, tx_frame_aborts=0 is_congested=0, mapped=1, is_isdn_d=0,
```

```
tx_limited=1 fast_if_number=15, fastsend=0x403339E4 map=0x7FFFFFFF, turbo_vector_name=Copperhead
```

```
to Draco switching lc_ip_turbo_fs=403A9EEC, lc_ip_mdfs=403A9EEC
```


Para CT3, debe obtener el `vc num`, que se puede obtener de la salida de **show interface serial CT3_interface_name**.

Ahora la información de CI se puede obtener de la consola SPA. En primer lugar, redirija el resultado de los comandos de la consola SPA al RP con el **comando spa_redirect rp ct3_freedm336**.

El comando **spa_ct3_test freedm show linkrec vc** muestra la información de CI necesaria.

dMFR

Depuración de Trayectoria de Control: **debug dmfr event**

Depuración de ruta de datos: **debug dmfr packets**

Depuración de errores de ruta de datos y ruta de control: **debug dmfr error**

Dumping de paquetes basados en CI: Consulte [dMLP](#).

dLFI

Depuración de Trayectoria de Control: **debug dlfi event**

Depuración de ruta de datos: **debug dlfi fragments**

Depuración de errores de ruta de datos y ruta de control: **debug dlfi error**

dDDR

No hay comandos especiales de depuración; debe utilizar [depuraciones dMLP](#).

En el caso de dLFIoLL, es posible que deban utilizarse las depuraciones dMLP y dLFI. Estas depuraciones no son condicionales y, por lo tanto, activarán todos los paquetes.

Preguntas Frecuentes

1. **¿Qué es dMLP?**dMLP es breve para PPP de links múltiples distribuidos (como se indica en [RFC1990](#)). Esta función es compatible con plataformas distribuidas, como las series 7500 y 7600 de Cisco. dMLP le permite combinar líneas T1/E1 (en un VIP en un Cisco 7500 Series Router o en un FlexWAN en un 7600 Series Router) en un paquete que tiene el ancho de banda combinado de varias líneas T1/E1. Esto permite a los clientes aumentar el ancho de banda más allá de T1/E1 sin necesidad de adquirir una línea T3/E3.
2. **¿Qué se "distribuye" en dMLP?**El término "distributed" implica que el switching de paquetes lo realiza el VIP y no el RSP. ¿Por qué? Las capacidades de switching de RSP son bastante limitadas y tiene muchos trabajos más importantes por hacer. El VIP capaz de conmutar paquetes descarga esta actividad del RSP. El IOS de Cisco basado en RSP todavía administra los links. La creación y eliminación de paquetes son realizadas por el RSP. Además, el RSP todavía realiza el procesamiento del plano de control PPP, incluida la gestión de todos los paquetes de control PPP (LCP, Autenticación y NCP). Sin embargo, una vez que se establece un paquete, la administración de los paquetes MLP se entrega al

VIP para que la CPU integrada conmute. El motor dMLP (en el VIP) gestiona todos los procedimientos MLP, incluidos la fragmentación, entrelazado, encapsulación, equilibrio de carga entre varios enlaces, y la ordenación y el reensamblado de fragmentos entrantes. Las funciones que realiza el VIP en un sistema 7500 las realiza Flexwan/Enhanced-FlexWAN en un sistema basado en 7600.

3. **¿Cómo puedo saber si el paquete está distribuido o no?**Ejecute el comando **show ppp multilink** en la consola del router:

```
Router# show ppp multilink
```

```
Multilink1, bundle name is udho2
Bundle up for 00:22:46
Bundle is Distributed
174466 lost fragments, 95613607 reordered, 129 unassigned
37803885 discarded, 37803879 lost received, 208/255 load
0x4D987C received sequence, 0x9A7504 sent sequence
Member links: 28 active, 0 inactive (max not set, min not set)
  Se11/1/0/27:0, since 00:22:46, no frags rcvd
  Se11/1/0/25:0, since 00:22:46, no frags rcvd
!--- Output suppressed.
```

4. **Si actualizo a RSP16 o SUP720, ¿mejorará mi rendimiento dMLP?**No. El rendimiento de switching de dMLP (o cualquier función distribuida) depende del VIP o de FlexWAN en cuestión. Por ejemplo, el rendimiento de un VIP6-80 será mejor que el de VIP2-50.
5. **¿Qué PA puedo utilizar con esta función?**PA-MC-T3PA-MC-2T3+PA-MC-E3PA-MC-2E1PA-MC-2T1PA-MC-4T1PA-MC-8T1PA-MC-8E1PA-MC-STM-1PA-MC-8TE1+PA-4T+PA-8TCT3IP-50 (sólo 7500)
6. **¿Cuántos links se pueden configurar en un solo paquete?**Hay muchas facetas en esta respuesta. El cuello de botella principal es la potencia de la CPU de la tarjeta de línea (VIP/FlexWAN/Enhanced-FlexWAN2). El límite de hardware es de 56 links por paquete, pero muchas veces no puede configurar esos muchos (y tiene tanta conmutación de tráfico), ya sea debido a la potencia de la CPU o a búferes limitados. Estos números se basan en esta directriz (en función de la CPU y la memoria en VIP/FlexWAN/Enhanced-FlexWAN2):VIP2-50 (con 4 MB de SRAM) T1s máx = 12VIP2-50 (con 8 MB de SRAM) T1s máx = 16VIP4-80 máx. T1s = 40VIP6-80 máx. T1s = 40FlexWAN max T1s = Se actualizará en breveEnhanced-FlexWAN max E1s = 21 E1s por bahía (total 42 E1s por tarjeta de línea)
7. **¿Hay algún cambio en el rendimiento si configuro 3 paquetes con 3 T1 cada uno o 1 paquete con 9 T1?**No hay cambios en el rendimiento, como se ha demostrado en las pruebas de laboratorio. Sin embargo, con un gran número de T1s en un solo paquete (digamos 24 o 28 T1s en un solo paquete), hay problemas con el agotamiento de las memorias intermedias. Se recomienda encarecidamente que no tenga más de 8 enlaces de miembro (T1/E1) en un solo paquete.
8. **¿Cómo se determina el ancho de banda de un paquete?**El ancho de banda de un paquete no se debe configurar. Es el ancho de banda agregado de todos los links de miembro. Si tiene 4 T1 en el paquete, el ancho de banda del paquete es de 6.144 Mbps.
9. **¿Cuál es mejor? ¿Equilibrio de carga CEF o dMLP?**No hay una respuesta sencilla a esto. Sus necesidades deciden cuál es mejor.**PROS de MLP:**El balanceo de carga CEF sólo se aplica al tráfico IP. MLP equilibra todo el tráfico enviado a través de un paquete.MLP mantiene el orden de los paquetes. La propia IP es tolerante con la reordenación, por lo que puede que no le importe; de hecho, el costo adicional que entraña el mantenimiento de la secuencia puede ser una razón para evitar el MLP. La IP está pensada para redes que pueden entregar datagramas de forma desordenada y todo lo que utilice IP se supone que puede lidiar con la reordenación. Sin embargo, a pesar de este hecho, la realidad es que la

reordenación todavía puede plantear un problema real. MLP proporciona una única conexión lógica al sistema de par. QoS se soporta en los paquetes de links múltiples. MLP proporciona capacidades de ancho de banda dinámico, ya que el usuario puede agregar o quitar enlaces de miembro según las necesidades actuales. MLP puede agrupar un mayor número de links, mientras que el balanceo de carga CEF está limitado a 6 trayectorias IP paralelas. El balanceo de carga CEF por flujo limita el ancho de banda máximo de cualquier flujo dado a un T1. Por ejemplo, los clientes que utilizan gateways de voz pueden tener muchas llamadas con el mismo origen y destino y, por lo tanto, utilizar sólo una ruta. **REP de MLP:** MLP agrega sobrecarga adicional a cada paquete o trama MLP utiliza una gran cantidad de CPU; dMLP requiere un uso intensivo de la CPU de la tarjeta de línea.

10. **¿Cómo puedo configurar varios paquetes entre dos routers?** Multilink determina a qué paquete se unirá un link en función del nombre del peer y del discriminador de terminales. Para crear varios paquetes distintos entre dos sistemas, el método estándar es forzar a algunos de los links a identificarse de forma diferente. El método recomendado es el uso del comando **ppp chap hostname *name name***.
11. **¿Puedo tener links de miembros de diferentes PA?** No. Si desea ejecutar dMLP, no se admite. Sin embargo, si los links de miembro se agregan desde diferentes PA, el control se da a RSP y ya no es dMLP. MLP todavía funciona, pero los beneficios del dMLP han desaparecido.
12. **¿Puedo combinar enlaces de miembros de ambas bahías?** No. Si desea ejecutar dMLP, no se admite. Sin embargo, si los links de miembro se agregan desde diferentes PA, el control se da a RSP y ya no es dMLP. MLP todavía funciona, pero los beneficios del dMLP han desaparecido.
13. **¿Puedo tener enlaces de miembros a través de diferentes VIP o FlexWAN?** No. Si desea ejecutar dMLP, no se admite. Sin embargo, si los links de miembro se agregan desde diferentes PA, el control se da a RSP y ya no es dMLP. MLP todavía funciona, pero los beneficios del dMLP han desaparecido.
14. **¿Puedo tener links de miembro a través de diferentes puertos desde un solo PA?** (Por ejemplo, un link miembro de cada puerto CT3 de un PA-MC-2T3+.) Yes. Mientras sea de la misma AP, no hay problemas.
15. **¿Puedo agrupar puertos T3 o E3?** No. Solo se permiten velocidades DS0, n*DS0, T1 y E1 con dMLP para 7500/VIP, 7600/FlexWAN y 7600/FlexWAN2. **Nota:** MLPPP distribuido se soporta solamente para links miembro configurados a velocidades T1/E1 o T1/E1 de velocidad inferior. Las interfaces STM-1/T3/T1 canalizadas también admiten dMLPPP a velocidades T1/E1 o T1/E1 de velocidad inferior. No se admite MLPPP distribuido para links de miembro configurados en T3/E3 de canal despejado o velocidades de interfaz superiores.
16. **¿Qué son los fragmentos "reordenados"?** Si el fragmento o paquete recibido no coincide con el número de secuencia esperado, el contador `reordenado` se incrementa. Para los tamaños de paquete variables, esto seguramente sucederá. Para los paquetes de tamaño fijo, esto también puede ocurrir porque el controlador PA procesa los paquetes que recibieron en un link y no pasa en base de ordenamiento cíclico (como se hace en dMLP mientras se transmiten los paquetes). Reordenar no significa pérdida de paquetes.
17. **¿Qué son los fragmentos "perdidos"?** Siempre que el fragmento o el paquete se recibe fuera de orden y descubre que se reciben fragmentos o paquetes fuera de orden en todos los links, el contador `fragmentos perdidos` se incrementa. Otro caso es cuando los fragmentos fuera de orden se almacenan en la lista y alcanza un límite (se decide en función de la SRAM en VIP y lo que sea que se asigne al paquete), el contador de

fragmentos perdidos se incrementa y se toma el siguiente número de secuencia en la lista para su procesamiento.

18. **¿Cómo detecta dMLP fragmentos perdidos?**Números de secuencia: Si está esperando a que llegue un fragmento con el número de secuencia N y todos los links reciben un fragmento con un número de secuencia superior a N, sabe que el fragmento N debe perderse, porque no hay forma de que pueda llegar legalmente detrás de fragmentos numerados más altos en el mismo link.timeout (tiempo de espera): Si se sienta a esperar demasiado tiempo por un fragmento, al final lo declarará perdido y seguirá adelante.Desbordamiento del búfer de montaje: Si está esperando a que llegue el fragmento N y mientras otros fragmentos (con números de secuencia superiores a N) están llegando en algunos de los links, entonces debe aparcar esos fragmentos en un búfer de reensamblado hasta que aparezca el fragmento N. Hay un límite en cuanto a la cantidad de memoria intermedia. Si el búfer se desborda, se declara nuevamente el fragmento N como perdido y se reanuda el procesamiento con lo que esté en el búfer.
19. **¿Qué se "ha perdido"?**Hay dos razones posibles para la pérdida de paquetes o fragmentos recibidos:Si el fragmento o paquete recibido está fuera de la ventana de rango de secuencia esperada, el paquete se descarta marcándolo como perdido recibido.Si el fragmento o paquete recibido está dentro de la ventana de rango de secuencia esperada, pero no puede asignar un encabezado de paquete para volver a propagar este paquete, el paquete se descarta y se marca como perdido recibido.
20. **¿Se admite el cifrado con dMLP?**No.
21. **¿Es compatible con la compresión de encabezado PFC?**No, no en la ruta distribuida. El router de extremo lejano no se recomienda configurar la compresión de encabezado PFC porque volvemos al modo no distribuido si recibimos tramas de encabezado comprimido o paquetes. Si desea continuar ejecutando dMLP, la compresión del encabezado PFC se debe inhabilitar en ambos extremos.
22. **¿Se admite la compresión de software con dMLP?**No, porque la compresión de software no funcionará en la ruta distribuida.
23. **¿Se admite la fragmentación en el lado de transmisión?**No con Vanilla dMLP. No hay problemas con la recepción de fragmentos con dMLP de vainilla, pero en el lado de transmisión, la fragmentación no ocurre. La fragmentación del lado de transmisión se soporta cuando **ppp multilink interleave** se configura en la interfaz dMLP.
24. **¿Podemos hacer ping a los links de miembro de un paquete MLP?**No, no puede configurar una dirección IP en los links de miembro.
25. **¿Hay alguna dependencia en el tamaño de los fragmentos de MTU y MLP del link?**No. El tamaño de MTU no tiene nada que ver con el tamaño del fragmento MLP, aparte de la restricción obvia de que un fragmento MLP, como cualquier otra trama, no puede exceder los tamaños de MTU de los links seriales.
26. **¿Es posible configurar dos paquetes MLP entre un solo par de routers?**Sí, es posible. Sin embargo, esto podría conducir a un equilibrio de carga dañado. Puede ser útil en testbed, para simular más de dos routers usando sólo dos routers, pero no tiene ningún valor real obvio.Todos los links que van a un peer común *deben* ser colocados en el mismo conjunto. Por definición, un conjunto es el conjunto de links que van a un par determinado.Un "peer" se identifica por los valores de Username y Endpoint Discriminator que ofrece durante las fases de LCP y autenticación. Si está intentando crear varios paquetes entre dos routers, significa que está intentando hacer que cada router se enmascare como más de un solo par a su contraparte. Deben identificarse adecuadamente.
27. **¿Pueden los links de miembros tener algoritmos de colocación en cola diferentes?**Todos

los mecanismos de colocación en cola relacionados con un paquete deben aplicarse en el nivel de agrupamiento y no en el nivel de link de miembro. Sin embargo, la configuración de un algoritmo de cola no debe afectar a la forma en que los paquetes se conmutan fuera del agrupamiento.

28. **¿Por qué se establece tx-queue-limit en 26 como valor predeterminado para los links de miembro para un paquete multilink cuando dMLP está habilitado en un Cisco 7500?** Para cualquier interfaz serial del ancho de banda T1/E1, el tx-queue-limit es alrededor de 4 o 5. Cuando agrupa T1s/E1s en links múltiples, el ancho de banda aumentaría para el paquete. Debido a que el switching se realizaría en función del ancho de banda de la interfaz MLP, necesita aumentar el límite de cola-tx de los links miembro. Sólo se utiliza uno de los links miembro, denominado link principal, para la conmutación, por lo tanto, su tx-queue-limit debe incrementarse. Además, este valor es empírico y se elige después de probar y, a continuación, ajustar a este valor. En general, las implementaciones no tienen más de 4 a 6 enlaces T1/E1 en un paquete. Un valor de 26 puede satisfacer perfectamente los 6 a 8 links T1/E1, por lo que se eligió este valor.
29. **¿Qué es el retraso diferencial y su valor en la implementación de dMLP?** dMLP admite un retraso diferencial de 30 ms. Eso significaría si se recibe un fragmento en una T y está fuera de servicio (esperando un número de secuencia 100, pero recibimos 101). Si la secuencia número 100 no se recibe hasta T+30 ms, 100 se declararía perdida y si puede comenzar el procesamiento desde 101, lo haría. En caso de que no pueda comenzar con 101 (si es un fragmento intermedio), buscará el fragmento que tiene el fragmento inicial (por ejemplo, 104) y comenzará desde allí.
30. **¿Qué sucede cuando los paquetes se fragmentan en el nivel IP con links múltiples en 7500?** Si los paquetes se fragmentan en el nivel IP, se transportan sin reensamblado en los saltos intermedios pero se reensamblan en el router de destino.
31. **¿Qué sucede cuando los paquetes se fragmentan en el nivel MLP en 7500?** Si los paquetes se fragmentan en el nivel MLP y si los paquetes reensamblados son mayores que MRU, los paquetes se descartan en links múltiples. La fragmentación del lado de transmisión se soporta en dMLP solamente con dLFI. Los paquetes se fragmentan en el nivel MLP solamente si el tamaño_de_paquete es mayor que el tamaño_frag y menor que la MRRU. Si se envían paquetes más que la MRU y si no se fragmenta en el nivel IP, el otro extremo descarta todo el tamaño de los paquetes que no se fragmentan en el nivel MLP porque los paquetes son más que la MRU.
32. **¿Cómo se calcula MRU?** La MRU se calcula según estas preferencias: Para los nuevos links de miembro que entran, MRRU se negocia nuevamente en el nivel LCP de acuerdo con la MRU configurada en los links de miembro. El valor configurado en la interfaz de link con el comando `ppp multilink mrru interface`. Si no se configura, el valor heredado de la configuración del comando `ppp multilink mrru` en la interfaz primaria. Si ambos valores están presentes, el valor de la interfaz de link tiene precedencia. La MRU predeterminada de 1524.

Mejoras de depuración

Estas mejoras se abordarán en el futuro. La planificación aún no se ha completado.

- Habilite el comando `debug frame-relay multilink` en la LC.
- Mejore las CLI de depuración actuales por interfaz y el número especificado de paquetes.

- Para dDDR, la funcionalidad de QoS todavía no se soporta. Esto solo se puede abordar con un argumento comercial adecuado.

Información Relacionada

- [Marcador CEF](#)
- [Configuración de DDR de igual a igual con perfiles de marcador](#)
- [MPLS: Soporte PPP Multilink](#)
- [Protocolo punto a punto multilink distribuido para routers de la serie Cisco 7500](#)
- [Distributed Multilink Frame Relay \(FRF.16\)](#)
- [Fragmentación y entrelazado de Link Distribuido sobre Líneas Arrendadas](#)
- [Voz sobre IP sobre links PPP con calidad de servicio \(LLQ/ prioridad IP RTP , LFI, cRTP\)](#)
- [Resolución de problemas de TechNotes: Cisco 7500 Series Router](#)
- [Página de soporte del producto del router - Cisco Systems](#)
- [Soporte Técnico y Documentación - Cisco Systems](#)