

# Roteador 75xx/76xx - Configurar e verificar recursos distribuídos

## Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Conventions](#)

[Recursos distribuídos](#)

[MLPPP distribuído](#)

[LFI distribuído](#)

[Diferenças entre dMLP e dLFloLL](#)

[MLFR distribuído](#)

[DDR distribuído](#)

[Recursos distribuídos pré-requisitos e restrições](#)

[Número de pacotes e links e requisitos de memória](#)

[Hardware e software MLPPP ou MLFR em placas de linha 7600 SIP](#)

[Vida útil de um pacote](#)

[Caminho De Dados Rx](#)

[Caminho de dados Tx](#)

[Reagrupamento](#)

[Configuração, verificação e depuração de recursos distribuídos](#)

[Configurando e verificando o dMFR](#)

[Configurando e verificando dMLP/dLFloLL](#)

[Configurando e verificando dLFloFR e dLFloATM](#)

[Configurando e verificando o dDDR](#)

[Depurando dMLP e dDDR](#)

[Perguntas mais freqüentes](#)

[Aprimoramentos de depuração](#)

[Informações Relacionadas](#)

## [Introduction](#)

Este documento ajuda você a entender, configurar e verificar estes recursos:

- Distributed Multilink Point to Point Protocol (dMLP)
- Fragmentação e intercalação de link distribuído (LFI)
- LFI distribuído sobre linha alugada (dLFloLL)
- LFI distribuído sobre Frame Relay (dLFloFR)

- LFI distribuído sobre ATM (dLFloATM)
- Diferenças entre MLP distribuído (dMLP) e dLFloLL
- Frame Relay Multilink distribuído (dMLFR)
- Roteamento de discagem por demanda (DDR) distribuído

## Prerequisites

### Requirements

Os leitores deste documento devem ter conhecimento dos recursos distribuídos para o Cisco 7500/7600/6500.

### Componentes Utilizados

As informações neste documento são baseadas nestas versões de software e hardware:

- Todas as plataformas Cisco 7500 e 7600 **Observação:** as informações neste documento também se aplicam às plataformas 6500.
- Versões relevantes do software Cisco IOS®, que esta tabela lista:

#### Suporte a recursos distribuídos para cada filial e plataforma

Recurso	Adaptadores de porta (PAs) <sup>1</sup> suportados	Plataformas 7500		Plataformas 7600	
		Principais versões do software Cisco IOS	Versões do Cisco IOS (Intercalação)	Principais versões do software Cisco IOS	Versões do software Cisco IOS (Intercalação)
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 <sup>2</sup>	12.0(3) T e posterior, 12.0(9) S e posterior	12.2SX 12.1E <sup>2</sup>	—
dLFloLL	Chan-PA PA-4T+ PA-8T	12.2T 12.3 12.3T 12.0S	12.2(8) T e posterior, 12.0(24) S e posterior	12,2SX	12.2(17) )SXB e posterior
dLFloFR	Chan-PA PA-4T+	12.2T 12.3 12.3T	12.2(4) T3 e	12,2SX	12.2(17) )SXB e

	PA-8T		posterior		posterior
<b>dLFlo ATM</b>	PA-A3 PA-A6	12.2T 12.3 12.3T	12.2(4) T3 e posterior	12,2SX	12.2(17) )SXB e posterior
<b>dMLFR</b>	Chan-PA PA-4T+ PA-8T	12,0S 12,3T	12.0(24) )S e posterior, 12.3(4) T e posterior	12,2SX	12.2(17) )SXB e posterior
<b>QoS em dMLP</b>	Chan-PA PA-4T+ PA-8T	12.0S 12.2T 12.3 12.3T	12.0(24) )S e posterior, 12.2(8) T e posterior	12,2SX	12.2(17) )SXB e posterior
<b>MPLS em dMLP MPLS em dLFlo LL</b>	Chan-PA PA-4T+ PA-8T	12,2T 12,3	12.2(15) )T11 e posterior, 12.3(5a) ) e posterior	12,2SX	12.2(17) )SXB e posterior
<b>DDR distribuído</b>	PA-MC- xT1 PA- MC-xE1 PA-MC- xTE1 PA- MCX- xTE1	12.3T	12.3(7) T e posterior	—	—

**Nota:** Esteja ciente destas informações:

1. Esses PAs suportam recursos distribuídos: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. O Cisco IOS Software Release 12.1E suporta esses recursos em plataformas 7500 e 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.

## [Conventions](#)

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

## Recursos distribuídos

Estes recursos são explicados neste documento:

- MLP distribuído
- LFI distribuído
- LFI distribuído sobre linha alugada
- LFI distribuído sobre Frame Relay
- LFI distribuído sobre ATM
- Diferenças entre dMLP e dLFIoLL
- MLFR distribuído
- Distributed Dialer
- Plataformas e placas de linha que suportam recursos distribuídos

### MLPPP distribuído

O recurso Distributed Multilink Point to Point Protocol (dMLP) permite combinar linhas T1/E1 completas ou fracionais em uma placa de linha (VIP, FlexWAN) em um roteador Cisco 7500 ou 7600 Series em um pacote que tem a largura de banda combinada de vários links. Você usa um pacote MLP distribuído para fazer isso. Um usuário pode escolher o número de pacotes em um roteador e o número de links por pacote. Isso permite que o usuário aumente a largura de banda dos links de rede além da de uma única linha T1/E1 sem a necessidade de comprar uma linha T3. Em não dMLP, todos os pacotes são comutados pelo Route Processor (RP); portanto, essa implementação afeta o desempenho do RP, com alta utilização da CPU para apenas algumas linhas T1/E1 executando MLP. Com o dMLP, o número total de pacotes que podem ser tratados no roteador é aumentado, pois o caminho de dados é tratado e limitado pela CPU e memória da placa de linha. O dMLP permite agrupar T1/E1 fracionário, a partir de DS0 (64 Kbps).

### LFI distribuído

O recurso dLFI suporta o transporte de tráfego em tempo real (como voz) e tráfego em tempo não real (como dados) em circuitos virtuais (VCs) de Frame Relay e ATM de baixa velocidade e em linhas alugadas sem causar retardo excessivo ao tráfego em tempo real.

Esse recurso é implementado com o uso de multilink PPP (MLP) sobre Frame Relay, ATM e linhas alugadas. O recurso fragmenta um pacote de dados grande em uma sequência de fragmentos menores, para permitir que pacotes em tempo real sensíveis a retardo e pacotes não em tempo real compartilhem o mesmo link. Os fragmentos são intercalados com os pacotes em tempo real. No lado receptor do link, os fragmentos são remontados e o pacote é reconstruído.

O recurso dLFI é frequentemente útil em redes que enviam tráfego em tempo real via Enfileiramento de baixa latência distribuído (como voz), mas têm problemas de largura de banda. Isso retarda o tráfego em tempo real devido ao transporte de pacotes de dados grandes e com menor tempo. Você pode usar o recurso dLFI nessas redes para desmontar os pacotes de dados grandes em vários segmentos. Os pacotes de tráfego em tempo real podem ser enviados entre esses segmentos dos pacotes de dados. Neste cenário, o tráfego em tempo real não sofre um retardo longo enquanto espera que os pacotes de dados de baixa prioridade atravessem a rede. Os pacotes de dados são remontados no lado receptor do link, de modo que os dados são entregues intactos.

O tamanho do fragmento do link é calculado com base no atraso do fragmento no pacote multilink, configurado com o comando **ppp multilink fragment-delay *n***, onde:

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

Esse tamanho de fragmento representa somente o payload IP. Não inclui os bytes de encapsulamento (tamanho do fragmento = peso - bytes de encapsulamento). Os termos "weight" e "fragment size" são vistos na saída do comando **show ppp multilink** no RP. Se o retardo de fragmento não estiver configurado, o tamanho padrão do fragmento será calculado para um retardo máximo de fragmento de 30.

**Observação:** com links de largura de banda variável, o tamanho do fragmento escolhido é baseado no link com a menor largura de banda.

### [LFI distribuído sobre linha alugada](#)

O recurso dLFloLL estende a funcionalidade de fragmentação e intercalação de links distribuídos para linhas alugadas. A LFI distribuída é configurada com o comando **ppp multilink interleave** na interface do grupo multilink. É aconselhável usar LFI distribuído em interfaces multilink com largura de banda inferior a 768 Kbps. Isso ocorre porque o atraso de serialização para pacotes de 1.500 bytes para largura de banda superior a 768 Kbps está dentro dos limites de atraso aceitáveis e não precisa ser fragmentado.

### [LFI distribuído sobre Frame Relay](#)

O recurso dLFloFR é uma extensão do recurso Multilink PPP sobre Frame Relay (MLPoFR). O MLP é usado para a fragmentação. Esse recurso é semelhante ao FRF.12, que suporta fragmentação e pode intercalar pacotes de alta prioridade via Enfileiramento de baixa latência.

O comando **ppp multilink interleave** no modelo virtual é necessário para ativar a intercalação na interface de acesso virtual associada. Além da ativação da comutação CEF distribuída na interface serial, este comando é um pré-requisito para que o LFI distribuído funcione.

**Observação:** a menos que você esteja usando o Frame Relay para internetworking ATM, recomenda-se usar FRF.12 em vez de dLFloFR, pois a utilização da largura de banda é melhor com FRF.12

### [LFI distribuído sobre ATM](#)

O recurso dLFloATM é uma extensão do recurso Multilink PPP over ATM (MLPoATM). O MLP é usado para a fragmentação.

O comando **ppp multilink interleave** no modelo virtual é necessário para ativar a intercalação na interface de acesso virtual associada. Além da ativação da comutação CEF distribuída na interface serial, este comando é um pré-requisito para que o LFI distribuído funcione.

Com dLFloATM, é muito importante que você selecione um tamanho de fragmento que faça com que os pacotes se encaixem em células ATM de forma que não provoquem preenchimento desnecessário nas células ATM. Por exemplo, se o tamanho do fragmento selecionado for 124 bytes, isso significa que um payload IP de 124 bytes finalmente passaria como 124 + 10 (Cabeçalho MLP) + 8 (Cabeçalho SNAP) = 142 bytes. É importante observar que o primeiro

fragmento sairia com  $124 + 10 + 2$  (tamanho do cabeçalho PID do primeiro fragmento) +  $8 = 144$  bytes. Isso significa que esse pacote usará três células ATM para transferir a carga e, portanto, usará a célula mais eficientemente embalada.

## Diferenças entre dMLP e dLFloLL

O dMLP não suporta fragmentação no lado de transmissão, enquanto o dLFloLL suporta.

**Observação:** a intercalação e a fragmentação usadas com mais de um link no pacote multilink para tráfego de voz não garantem que o tráfego de voz recebido através de vários links no pacote será recebido em ordem. A ordenação correta de voz é tratada nas camadas superiores.

## MLFR distribuído

O recurso MLFR distribuído introduz a funcionalidade baseada no Frame Relay Forum Multilink Frame Relay UNI/NNI Implementation Agreement (FRF.16) para os roteadores das séries Cisco 7500 e 7600 habilitados para placa de linha. O recurso MLFR distribuído oferece uma maneira econômica de aumentar a largura de banda para aplicativos específicos, pois permite que vários links seriais sejam agregados em um único pacote de largura de banda. O MLFR é suportado em interfaces de usuário para rede (UNI) e interfaces de rede para rede (NNI) em redes Frame Relay.

O pacote é composto de vários links seriais, chamados de links de pacote. Cada link de pacote dentro de um pacote corresponde a uma interface física. Os links do pacote são invisíveis para a camada de enlace de dados do Frame Relay, de modo que a funcionalidade do Frame Relay não pode ser configurada nessas interfaces. A funcionalidade regular do Frame Relay que você deseja aplicar a esses links deve ser configurada na interface do pacote. Os links do pacote são visíveis para dispositivos pares.

## DDR distribuído

O recurso DDR distribuído permite a comutação distribuída nas interfaces do discador. Sem esse recurso, todo o tráfego de discagem deve ser direcionado ao host para comutação. Com ele, somente pacotes de controle são enviados ao RP, enquanto a decisão de switching é feita nos próprios VIPs depois que uma conexão é estabelecida.

A configuração do Legacy Dialer e a configuração do Dialer Profile são suportadas somente com o encapsulamento PPP. O MLP nas interfaces do discador também é suportado. A QoS não é suportada com switching distribuída nas interfaces do discador.

## Recursos distribuídos pré-requisitos e restrições

### Prerequisites

Estes são pré-requisitos gerais para todos esses recursos distribuídos:

- O switching distribuído do Cisco Express Forwarding (dCEF) deve ser ativado globalmente.
- A comutação dCEF deve ser ativada na interface serial do membro, que faz parte do pacote MLP.
- A comutação dCEF deve ser ativada no link físico das interfaces dLFloFR e dLFloATM.
- A configuração de intercalação é necessária para distribuir LFloFR e LFloATM.

- Configure a largura de banda necessária na interface de modelo virtual para as interfaces dLFloFR e dLFloATM.
- Quando as depurações PPP são ativadas no RP, você pode observar o MLP: Encaminhado para mensagem de interface incorreta no Route Switch Processor (RSP). Como essa mensagem é confusa e indesejada, especialmente se a mensagem for para pacotes do Cisco Discovery Protocol (CDP), você deve configurar **nenhum cdp ativado** nos links de membros do pacote.
- Todos os links de membros do pacote devem ter o keepalive ativado.

## Restrições

Essas são restrições gerais para todos esses recursos distribuídos:

- As linhas T1 e E1 não podem ser misturadas em um pacote.
- O atraso diferencial máximo suportado é de 30 ms.
- Todas as linhas em um pacote devem residir no mesmo adaptador de porta (PA).
- Não há suporte para compactação de hardware.
- VIP ou FlexWAN CEF é limitado somente a IP; todos os outros protocolos são enviados ao RSP.
- A fragmentação não é suportada no lado de transmissão para dMLP e dMLFR.
- Muitos dos mecanismos de enfileiramento mais antigos não são suportados pelo dLFI. Esses mecanismos incluem: Enfileiramento justo em uma interface de modelo virtual Detecção aleatória em uma interface de modelo virtual Enfileiramento personalizado Enfileiramento de prioridade
- Enfileiramento justo, detecção aleatória (dWRED) e enfileiramento de prioridade podem ser configurados em uma política de tráfego com a CLI de QoS modular.
- Somente um link por pacote MLP é suportado quando você está usando dLFloFR ou dLFloATM. Se mais de um link for usado em um pacote MLP ao usar dLFloFR ou dLFloATM, o dLFI será desativado automaticamente. Ao usar dLFI em linhas alugadas, mais de um link pode ser configurado com dLFI no pacote MLP.
- Com dLFloATM, somente aal5snap e aal5mux são suportados. O encapsulamento aal5nlpid e aal5ciscopp não são suportados.
- Somente voz sobre IP é suportada; Voz sobre Frame Relay e Voz sobre ATM não são suportados pelo recurso dLFI.
- A configuração do Compressed Real-Time Protocol (CRTP) não deve ser configurada na interface multilink quando você usa esta combinação de recursos: Interface multilink com LFI ativado O pacote multilink tem mais de um link de membro A política de QoS com recurso de prioridade está habilitada na interface multilink

Com a configuração dMLP e dLFI, os pacotes de prioridade não carregam o cabeçalho MLP e o número de sequência, e o MLP distribuirá os pacotes de prioridade em todos os links de membros. Como resultado, os pacotes compactados pelo CRTP podem chegar fora de ordem no roteador receptor. Isso proíbe que o CRTP descompacte o cabeçalho do pacote e força o CRTP a descartar os pacotes.

## Recomendações

Recomenda-se que os links de membros em um pacote tenham a mesma largura de banda. Se você adicionar links de largura de banda desiguais ao pacote, isso levará a uma grande reordenação de pacotes, o que fará com que o throughput global do pacote diminua.

Recomenda-se o uso de VIP2-50 (com 8 MB de SRAM) ou superior com esses recursos distribuídos.

## [Número de pacotes e links e requisitos de memória](#)

Consulte [Distributed Multilink Point-to-Point Protocol para Cisco 7500 Series Routers](#).

## [Hardware e software MLPPP ou MLFR em placas de linha 7600 SIP](#)

O MLP e o MLFR podem ser baseados em software ou hardware. No MLP ou MLFR baseado em hardware, o Freedm fornece a funcionalidade multilink e os cabeçalhos MLP são adicionados pelo Freedm Chip. No MLP ou MLFR baseado em software, a CPU da placa de linha SIP fornece a funcionalidade multilink (que é semelhante às implementações VIP e FlexWAN).

Essas são as limitações e condições para executar MLP ou MLFR baseado em hardware.

- Só pode haver um máximo de 336 pacotes por placa de linha e 168 pacotes por Avaliação de Postura de Segurança (SPA - Security Posture Assessment) (Freedm).
- Só pode haver um máximo de 12 DS1/E1 por pacote.
- Todos os links devem pertencer ao mesmo SPA (Freedm).
- Todos os links no pacote devem operar na mesma velocidade.
- O tamanho do fragmento TX pode ser 128, 256 ou 512. Um tamanho de fragmento configurado pela CLI é mapeado para o tamanho de fragmento suportado mais próximo.

```
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
```
- O tamanho do fragmento RX pode ser de 1 a 9,6 KB.
- O formato proprietário da Cisco não pode ser suportado (MLFR).

No hardware LFI, se houver apenas um link no pacote e se esse for DS1/E1, a fragmentação e a intercalação serão feitas pelo Freedm.

A saída de **show ppp multilink** mostra se a implementação de hardware está em execução.

```
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.
```

Se multilink for baseado em software, a saída **show ppp multilink não terá** Multilink em Hardware na saída.

## [Vida útil de um pacote](#)



## Caminho De Dados Rx

1. Pacote recebido pelo driver.
2. O encapsulamento está verificado: a seguir Encapsulamento básico: Em dMLP, o tipo de encapsulamento para a interface de entrada é ET\_PPP. Em dMLFR, o tipo de encapsulamento para a interface de entrada é ET\_FRAME\_RELAY. Em dLFloLL, o tipo de encapsulamento para a interface de entrada é ET\_PPP. Em dLFloFR, o tipo de encapsulamento para a interface de entrada é ET\_FRAME\_RELAY. Em dLFloATM, o tipo de encapsulamento para a interface de entrada é ET\_ATM. No dDialer, o tipo de encapsulamento é ET\_PPP. Processamento de encapsulamento adicional: Para ET\_PPP, o NLPID é detectado. Para o dMLP, o NLPID é MULTILINK. Para o dLFloLL, há duas coisas a considerar: Pacotes VoIP—Eles não têm um cabeçalho MLP e têm um NLPID que indica IP. Pacotes de dados—O NLPID é MULTILINK. Para o dDialer, os pacotes não terão um cabeçalho MLP e terão um NLPID que indique IP. **Observação:** nesse caso, você pode configurar o dCRTP (Distributed Compressed Real-Time Protocol). Em caso afirmativo, o cabeçalho é descompactado antes do processamento posterior.
3. Para ET\_FRAME\_RELAY, se o link no qual o pacote é recebido estiver configurado para dMLFR, o pacote será processado para dMLFR
4. Para dLFloFR e dLFloATM, o tipo de encapsulamento é ET\_FRAME\_RELAY e ET\_ATM, respectivamente; mas dentro disso há um cabeçalho PPP. O cabeçalho PPP, como com dLFloLL, indicará se o pacote é um pacote de voz ou um pacote de dados. Se dCRTP estiver configurado, o cabeçalho será descompactado antes do processamento posterior. Os pacotes de voz são comutados imediatamente. Um pacote de dados fragmentado terá que ser remontado antes de ser comutado. Com o ET\_PPP, você pode encontrar pacotes de link PPP; e com ET\_FRAME\_RELAY, você pode encontrar pacotes de controle MLFR. Todos esses pacotes de controle são encaminhados ao RP para processamento.
5. Dependendo da decodificação mencionada acima, o pacote é verificado quanto ao tipo de comutação necessária. O tipo de link determinará se o pacote deve ser comutado por IP ou comutado por MPLS. Os pacotes são então fornecidos às respectivas funções de comutação.
6. Com o agrupamento em conjunto com recursos distribuídos, o vetor de switching turbo fast IP é roubado. Isso é feito porque o pacote é recebido no link do membro; no entanto, ele deve ser tratado de forma que seja recebido no pacote. Você também precisa verificar se há pacotes de controle que são direcionados ao host. Principalmente em dMLFR, há pacotes da Local Management Interface (LMI) que *não* são pacotes de controle MLFR. Para eles, é usada uma parte diferente do espaço numérico do dLCI. Sempre que o dLCI é decodificado para cair nesse espaço, o pacote é direcionado para o host, porque é reconhecido como um pacote LMI. Os pacotes VoIP (enfileirados na Fila de baixa latência) são simplesmente comutados sem a adição do cabeçalho MLP. Os recursos distribuídos podem receber e reagrupar pacotes quando pacotes de dados fragmentados são recebidos. O processo de remontagem é explicado em uma seção posterior. Se o pacote precisar ser comutado por tag, ele será passado para a rotina de tag-switching, em dMLP. Caso contrário, se for comutado por IP, ele será passado para a rotina de comutação por IP. **Observação:** todos os pacotes não IP são direcionados ao host, em dMLFR.
7. **IP:** A função de comutação IP é comum a todos os pacotes. Ele faz principalmente três coisas: Faça o processamento necessário dos pacotes, caso haja algum recurso configurado. Além disso, quando o Distributed Dialer for usado, faça as atualizações do temporizador de ociosidade aqui quando um "pacote interessante" for recebido. Consulte

[dialer idle-timeout \(interface\)](#) , [dialer fast-idle \(interface\)](#) e [Configuring a Dialer Profile](#) para obter detalhes do parâmetro de configuração do **temporizador ocioso**. Nos roteadores 75xx, a adjacência indicará a `tx_acc_ptr` para a interface de saída. Se a interface de saída for uma interface de acesso virtual, a `tx_acc_ptr` será `NULL`. Nesse caso, corrija o encapsulamento e obtenha o `tx_acc_ptr` do `fib hwidb`. Essa correção de pesquisa e encapsulamento é necessária em `dLFIoFR` e `dLFIoATM`. Em `dLFIoLL`, o link é tratado como parte de um pacote Multilink. **Observação:** o TTL do pacote é ajustado aqui e a verificação de fragmentação de IP é feita. O `mci_status` está definido como `RXTYPE_DODIP` para todos os pacotes.

- Com a decisão de switching tomada, o pacote está pronto para ser enviado da interface. A interface é verificada para determinar se suporta switching local. Em caso afirmativo, ele é enviado diretamente por meio do `fastsend`. Caso contrário, é feita uma tentativa de comutar o cache de rota. Observe que, caso a QoS esteja configurada para a interface, o vetor de switching local é roubado pela QoS. O HQF enfileirará o pacote e processará ainda mais o pacote, antes que ele seja finalmente enviado para fora da interface. Esse é o caso do `dLFI`. Para o `dLFI`, a fragmentação e a intercalação são definidas. O QoS trata a invocação de nossa rotina de fragmentação e intercala os pacotes fragmentados com os pacotes de voz que serão enfileirados na fila de prioridade (se o LLQ estiver configurado). Isso garante que os pacotes VoIP não sofram o atraso necessário para enviar pacotes de dados enormes pelo link.

## Caminho de dados Tx

O `vip_dtq_consumer` recebe o pacote e recebe o número da interface, do qual recebe o `idb`. A rotina `fastsend` que corresponde ao `idb` é chamada:

### i) Envio rápido

- Em `dMFR`, a estrutura `fr_info` é recuperada da tabela que corresponde a `if_index` ao `fr_info`. Os pacotes de controle são enviados. O cabeçalho do quadro fornecerá o `dLCI`, que o ajudará a determinar se este é um pacote LMI ou um pacote de dados. O campo `dLCI` no cabeçalho do quadro é substituído pelo número de sequência `dmfr`. Números de sequência separados são usados para LMI e pacotes de dados. **Observação:** números de sequência separados são usados para `dLCIs` separados.
- Em `dMLP`, os pacotes de controle são enviados com prioridade definida como alta. Com pacotes de dados, se `dCRTP` estiver configurado, o cabeçalho será compactado. O cabeçalho VIP MLP que inclui as informações de sequência é adicionado e enviado dos links de membros.
- Em `dLFI`, o HQF intercepta os pacotes a serem enviados através da interface. Se for um pacote de voz, o pacote de voz será colocado na fila de prioridade (se o LLQ estiver configurado) e será enviado para fora da interface sem o encapsulamento MLP. Com pacotes de dados, ele chama o código de fragmentação `dLFI`, que retorna fragmentos para o código QoS, que são intercalados com o tráfego de prioridade para que os requisitos de atraso do tráfego de voz sejam atendidos. Além disso, se `dCRTP` estiver configurado, somente o cabeçalho do pacote de voz será compactado. Os cabeçalhos dos pacotes de dados são deixados como estão.
- No `dDialer`, o pacote é classificado para redefinir o temporizador de ociosidade do link de saída antes que o pacote seja enviado. Isso é feito depois que o link de saída é escolhido, caso vários links sejam vinculados ao mesmo discador. Nenhum cabeçalho é adicionado

aos pacotes do discador. Assim, o sequenciamento e a remontagem de pacotes não são suportados nas interfaces do Discador.

**Observação:** em dMLP, dDialer, dMLFR e dLFI com vários links, o link físico no qual o tráfego é encaminhado depende do congestionamento do link. Se o link estiver congestionado, vá para o próximo link e assim por diante. (dMLFR, dMLP sem QoS e recursos do dDialer também escolhem links com base no número de bytes colocados no link. Ele escolhe o próximo link, se o link atual já tiver transmitido sua cota de bytes, em uma base de rodízio. Essa cota é decidida por frag\_bytes para o link. Para interfaces de membro do discador, frag\_bytes é definido para o valor padrão da largura de banda da interface.)

**Observação:** nas configurações de HQF nas interfaces do VIP de saída, o HQF rouba o vetor dtq\_consumer. O pacote DMA para o VIP de saída primeiro passa pela verificação de HQF. Se a QoS estiver configurada na interface de saída, o HQF entra para processar o pacote, antes que o pacote seja enviado rapidamente para fora da interface.

## Reagrupamento

As interfaces dDialer simples não suportam remontagem e sequenciamento. Para ativar isso nas interfaces do discador, o MLP sobre interfaces do discador precisará ser configurado. Se isso for feito, o caminho Rx e Tx serão idênticos aos caminhos dMLP. Quando os pacotes são recebidos, o número de sequência é comparado ao número de sequência esperado.

- Se os números de sequência corresponderem: Se o pacote for um pacote não fragmentado, a remontagem não será necessária. Continue com as etapas de comutação adicionais. Se o pacote for um fragmento, verifique os bits de início e fim e construa o pacote como e quando os fragmentos forem recebidos.
- Se os números de sequência não corresponderem: Se o número de sequência estiver dentro da janela esperada de números de sequência, coloque-o na "lista de fragmentos não atribuídos" classificada. Mais tarde, quando um número de sequência esperado não é recebido, essa lista é verificada, caso o pacote tenha sido armazenado aqui. Se o número de sequência não estiver na janela, descarte-o e informe "fragmento perdido recebido". Se um tempo limite ocorrer mais tarde enquanto espera por esse pacote, o receptor é sincronizado novamente e começa novamente com o próximo pacote recebido.

Em todos esses casos, um fluxo de pacote ordenado corretamente é enviado para fora dessa interface. Se forem recebidos fragmentos, um pacote completo é formado e enviado.

## Configuração, verificação e depuração de recursos distribuídos

Esta seção aborda os comandos **show** e **debug** disponíveis para verificar e depurar cada um dos recursos distribuídos.

### Configurando e verificando o dMFR

#### Configuração de exemplo de 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
```

**Observação:** a interface MFR é como outra interface FR e, portanto, suporta a maior parte da configuração 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 o status do pacote MFR no 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
```

Isso indica que duas interfaces foram adicionadas corretamente e uma interface ainda não negociou as mensagens LIP MLFR.

Para obter mais informações sobre o pacote MFR e links de membro, emita 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 depuração MFR](#)

Essas depurações são úteis para solucionar problemas em que um link não é adicionado ao pacote.

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

**Observação:** quando uma interface MFR ou interface serial específica não é especificada, isso ativa a depuração para todos os links MFR. Isso pode ser impressionante se o roteador tiver um grande número de links MFR.

Para depurar pacotes MFR recebidos no RP, bem como para depurar as atividades de controle MFR, esta depuração é útil:

```
debug frame-relay multilink
```

**Observação:** sob tráfego intenso, isso pode sobrecarregar a CPU.

## [Verifique o status do pacote dMLFR no LC](#)

```
show frame-relay multilink
```

**Observação:** no momento, isso não está disponível no LC, mas será adicionado em breve. Até lá, 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

```

## Configurando e verificando dMLP/dLFIoLL

### Configuração PPP Multilink

```

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

```

Exemplo de configuração na interface 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
 !

```

**Observação:** o comando **ppp chap hostname M1** não significa realmente que a autenticação CHAP esteja habilitada. A string **M1** neste comando atua como discriminador de ponto final e só é necessária se houver mais de um pacote multilink entre os mesmos dois roteadores. Nesse caso, todos os links que pertencem a um pacote devem ter o mesmo discriminador de endpoint e

nenhum dois links que pertençam a um pacote diferente devem ter o mesmo discriminador de endpoint.

## [Parâmetros de configuração opcionais](#)

### [no] ppp multilink interleave

Isso permite a intercalação no pacote multilink. Isso funciona em conjunto com CLI QoS modular. Pacotes de alta prioridade serão transmitidos sem a adição da sequência e do cabeçalho MLP, enquanto outros pacotes serão fragmentados e transmitidos com a sequência e o cabeçalho MLP.

**Observação:** quando a intercalação é habilitada com mais de um link, é possível que o tráfego de alta prioridade seja reordenado. Quando a intercalação está ativada ou desativada, é necessário redefinir o pacote para ativá-lo na placa de linha.

```
ppp multilink mrru local value
```

Especifica a unidade de recepção máxima no multilink; os pacotes até esse tamanho serão aceitos pela interface multilink. O tamanho aqui exclui o cabeçalho MLP.

```
ppp multilink mrru remote value
```

Especifica o MRRU mínimo que a extremidade remota deve suportar. Se a MRRU de extremidade remota for menor que esse valor, a negociação do pacote falhará.

```
ppp multilink fragment delay seconds
```

Especifica o atraso permitido em milissegundos (ms) causado por um fragmento de dados. Em outras palavras, o valor de atraso é usado para calcular o tamanho máximo de fragmento permitido. A implementação distribuída difere da implementação do Cisco IOS das seguintes maneiras:

1. A fragmentação não é executada a menos que a intercalação esteja habilitada.
2. Com links de largura de banda variável, o tamanho do fragmento escolhido é baseado na interface de largura de banda mínima.

```
ppp multilink fragment disable
```

Esse comando não adiciona nenhuma funcionalidade na implementação distribuída. A fragmentação só ocorre quando a intercalação está ativada; e, quando a intercalação está ativada, o comando **ppp multilink fragment disable** é ignorado.

## [Verifique o status do pacote dMLP no 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. Quando o pacote está no modo distribuído, isso é exibido na saída de **show ppp multilink**:  
o pacote é distribuído. Caso contrário, o pacote por algum motivo não é distribuído.
2. Quando o **ppp multilink interleave** é configurado e ativado na placa de linha, a saída **show ppp multilink** inclui as estatísticas de dLFI em que:  
Fragmentado — Indica a contagem de fragmentos que foram transmitidos e recebidos.  
Desfragmentado — Indica a contagem de pacotes que foram transmitidos ou recebidos sem serem fragmentados.  
Remontado — Indica o número de pacotes completos que foram remontados. Quando a intercalação não está ativada, a saída é semelhante a esta:

```
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
```

O tamanho do fragmento no exemplo anterior é 760 bytes.

## [Verifique o status do pacote dMLP no 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

Com o dMFR, os números de sequência são mantidos por dLCI, com o número de sequência no pacote usado para o dLCI da LMI.

Campo	Descrição
dmlp_ipc_config_count	Número de mensagens IPC recebidas pelo LC para configuração de multilink ou MLFR
dmlp_bundle_count	Número de pacotes MLP e MLFR no LC
dmlp_ipc_failure_count	Número de mensagens de configuração que resultaram em falha no LC. Deve ser 0; se for diferente de zero, pode haver um problema.
vetores de etiquetas	Indica o idb para tag_Optimal_fs e o idb para os vetores ip2tag_Optimal_fs usados na comutação de tags.
board_encap	Indica o vetor board_encap usado para adicionar 2 bytes de encapsulamento de placa, se houver links canalizados em uma plataforma 7500. Deve ser

	NULL, se o link contiver interfaces não canalizadas.
máx. de partículas	Número máximo de partículas que podem ser mantidas no buffer de remontagem
mrru	O tamanho máximo do pacote aceito sem considerar o encapsulamento MLP. Não aplicável para a interface MLFR.
seq_window_size	O tamanho máximo da janela para os números de sequência
working_pak	Indica o pacote atual em remontagem. NULL, se não.
working_pak_cache	Aponte para o pacote estático usado para remontagem. Isso é alocado quando o primeiro pacote não completo é recebido pelo pacote.
una_frag_list	Primeira entrada na fila de remontagem. Se a entrada não for NULL e não for alterada, ela indica que o temporizador não está executando um problema de software.
una_frag_end	Última entrada na fila de remontagem
rcved_end_bit	Indica que o pacote recebeu um bit final, portanto está procurando um bit inicial.
is_loss_frag	É verdade, se um fragmento for declarado perdido. Isso é limpo quando um fragmento com a sequência esperada é recebido.
resync_count	Indica o número de vezes que o receptor estava fora de sincronia com o transmissor e teve que resincronizar iniciando com o último fragmento sequenciado recebido.
timeout	Indica que o tempo limite de remontagem ocorreu e que os pacotes estão sendo processados da fila de remontagem.
timer_start	Número de vezes que o temporizador de remontagem foi iniciado
timer_running	Indica se o temporizador de remontagem está ou não em execução.

temporizador_count	Indica o número de vezes que o temporizador de remontagem expirou.
next_xmit_link	O link no qual o próximo pacote será transmitido
Membro	Campo de bit que indica os membros presentes.
Congestionamento	Campo não usado em todas as filiais. Indica quais links de membros não estão congestionados.
dmlp_orig_pak_to_host	O vetor usado para enviar pacotes ao RP.
dmlp_orig_fastsend	O driver original envia rapidamente antes que o MLP ou o MLFR modificassem o envio rápido do driver.
fragmentos perdidos	Número de fragmentos perdidos (o receptor não recebeu esses fragmentos). Isso é limpo periodicamente quando uma atualização é enviada ao host.
Reordenado	Número de fragmentos recebidos fora da ordem esperada. Isso é limpo periodicamente quando uma atualização é enviada ao host.
Descartado	Número de fragmentos descartados porque não foi possível criar um pacote completo
perdido recebido	Número de fragmentos recebidos que se pensou terem sido perdidos. Isso indica que o atraso entre links é maior que o tempo limite de remontagem dMLP de 30 ms.

## [Configurando e verificando dLFloFR e dLFloATM](#)

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

```
policy-map llq
  class voip
    priority
```

```
int virtual-templatel
  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
```

## [Verificar o status do pacote dLFIoFR/ATM no 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**

**Observação:** o pacote será distribuído somente quando **ppp multilink interleave** estiver configurado no modelo virtual; sem esse comando, o pacote não será distribuído.

## [Verificar o status do pacote dLFIoFR/ATM no LC](#)

Para verificar se o dLFI está realmente funcionando corretamente no LC, emita 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

```

Deve haver uma camada prioritária e uma camada WFQ. A fragmentação será feita no blt da camada de folha do WFQ.

## [Configurando e verificando o dDDR](#)

O DDR distribuído é ativado quando você habilita o **ip cef distribuído** na configuração global e o **ip route-cache distribuído** nas interfaces do discador.

```

!--- 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

```

Não há outras configurações especiais para DDR distribuído. A configuração adicional segue a configuração normal do DDR.

## [Verificar o roteamento de discagem distribuída sob 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: 0x807FFFFFFF 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
```

O tipo de discador informa o tipo de discador usado. `ISDN` implica configuração de discador herdado e `PROFILE` implica configuração de perfil de discador. O estado do discador indica o estado atual do discador. O estado de uma interface de discador desconectada será `ocioso`. O temporizador de inatividade é redefinido sempre que o tráfego interessante é visto. Se esse temporizador expirar, a interface será desconectada imediatamente. O temporizador `ocioso` é um parâmetro configurável. Para obter mais informações, consulte [Configuração de DDR Ponto-a-Ponto com Perfis de Discador](#).

#### **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
```

As variáveis mostradas são as mesmas para dMLP.

## [Depurando dMLP e dDDR](#)

### [Depurações disponíveis no RP](#)

#### **dDDR**

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

Emita este comando para depurar funções de caminho de controle, como configuração de chamada e assim por diante. Para obter mais informações, consulte [debug dialer events](#) .

```
debug ip cef dialer
```

Emita este comando para depurar eventos de discador relacionados ao CEF. Para obter mais

informações, consulte [CEF do discador](#).

## Depurações disponíveis no LC

### dMLP

Depuração de caminho de controle: **debug multilink event**

Depuração do caminho de dados: **debug multilink fragments**

Caminho de dados e depuração de erro de caminho de controle: **debug multilink error**

### Depuração de dMLP em placas de linha SIP

Despejando pacotes com base em CI: Pacotes de dados e pacotes de controle podem ser despejados em placas de linha com base na CI de controle e na CI de sequência.

**test hw-module *subslot\_num* dump ci *CI-NUM* [rx|tx] *num\_packets\_to\_dump***

As ICs podem ser obtidas desta forma:

```
!--- 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, você deve obter o número *vc*, que pode ser obtido da saída de **show interface serial *CT3\_interface\_name***.

Agora, as informações de CI podem ser obtidas do console SPA. Primeiro, redirecione a saída dos comandos do console SPA para o RP com o comando **spa\_redirect rp ct3\_freedm336**.

O comando **spa\_ct3\_test freedm show linkrec *vc* mostra as informações de CI necessárias**.



## dMFR

Depuração de caminho de controle: **debug dmfr event**

Depuração do caminho de dados: **debug dmfr packets**

Caminho de dados e depuração de erro de caminho de controle: **debug dmfr error**

Despejando pacotes com base em CI: Veja [dMLP](#).

## dLFI

Depuração de caminho de controle: **debug dlfi event**

Depuração do caminho de dados: **debug dlfi fragments**

Caminho de dados e depuração de erro de caminho de controle: **debug dlfi error**

## dDDR

Não há nenhum comando de depuração especial; você deve usar [depurações dMLP](#).

No caso de dLFloLL, as depurações dMLP e dLFI podem precisar ser usadas. Essas depurações não são condicionais e, portanto, dispararão para todos os pacotes.

## Perguntas mais freqüentes

1. **O que é dMLP?**dMLP é curto para PPP Multilink Distribuído (conforme declarado em [RFC1990](#) ). Esse recurso é suportado por plataformas distribuídas, como as séries Cisco 7500 e 7600. O dMLP permite combinar linhas T1/E1—em um VIP em um roteador da série Cisco 7500 ou um FlexWAN em um roteador da série 7600—em um pacote que tem a largura de banda combinada de várias linhas T1/E1. Isso permite que os clientes aumentem a largura de banda além de T1/E1 sem a necessidade de comprar uma linha T3/E3.
2. **O que é "distribuído" no dMLP?**O termo "distribuído" implica que a comutação de pacotes é feita pelo VIP e não pelo RSP. Por quê? Os recursos de comutação de RSP são bastante limitados e têm muito mais trabalho a ser feito. O VIP capaz de comutar pacotes descarrega essa atividade do RSP. O Cisco IOS baseado em RSP ainda gerencia os links. A criação e o descarte do pacote são feitos pelo RSP. Além disso, o processamento do plano de controle do PPP ainda é feito pelo RSP, incluindo o tratamento de todos os pacotes de controle do PPP (LCP, Autenticação e NCPs). No entanto, quando um pacote é estabelecido, o tratamento dos pacotes MLP é convertido para o VIP para comutação pela CPU integrada. O mecanismo dMLP (no VIP) lida com todos os procedimentos de MLP, incluindo fragmentação, intercalação, encapsulamento, balanceamento de carga entre vários enlaces e ordenação e remontagem de fragmentos de entrada. As funções feitas pelo VIP em um sistema 7500 são feitas pelo Flexwan/Enhanced-FlexWAN em um sistema baseado em 7600.
3. **Como saber se o pacote está distribuído ou não?**Emita o comando **show ppp multilink** no console do roteador:

```
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. **Se eu atualizar para RSP16 ou SUP720, meu desempenho de dMLP será melhor?** Não. O desempenho de comutação do dMLP (ou qualquer recurso distribuído) depende do VIP ou FlexWAN em questão. Por exemplo, o desempenho de um VIP6-80 será melhor do que o desempenho com VIP2-50.
5. **Quais PAs posso usar com esse recurso?** 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 (somente 7500)
6. **Quantos links podem ser configurados em um único pacote?** Há muitas facetas nesta resposta. O principal gargalo é a potência da CPU da placa de linha (VIP/FlexWAN/Enhanced-FlexWAN2). O limite rígido é de 56 links por pacote, mas muitas vezes você não pode configurar esses muitos (e tem essa quantidade de comutação de tráfego), devido à potência da CPU ou buffers limitados. Esses números são baseados nesta diretriz (baseada na CPU e na memória no VIP/FlexWAN/Enhanced-FlexWAN2): VIP2-50 (com 4 MB de SRAM) máx. T1s = 12VIP2-50 (c/ SRAM de 8 MB) máx. T1s = 16VIP4-80 máx. T1s = 40VIP6-80 máx. T1s = 40FlexWAN máx. T1s = será atualizado em breveEnhanced-FlexWAN máx E1s = 21 E1s por compartimento (42 E1s agregados por placa de linha)
7. **Há uma mudança no desempenho se eu configurar 3 pacotes com 3 T1s cada ou 1 pacote com 9 T1s?** Não há mudança no desempenho, como comprovado em testes de laboratório. No entanto, com um grande número de T1s em um único pacote (por exemplo, 24 ou 28 T1s em um único pacote), há problemas com o esgotamento dos buffers. É altamente recomendado que você não tenha mais de 8 links de membros (T1/E1) em um único pacote.
8. **Como a largura de banda de um pacote é determinada?** A largura de banda de um pacote não deve ser configurada. É a largura de banda agregada de todos os links membros. Se você tiver 4 T1s no pacote, a largura de banda do pacote será de 6,144 Mbps.
9. **Qual é melhor? Balanceamento de carga CEF ou dMLP?** Não há uma resposta simples para isso. Suas necessidades decidem qual é melhor. **PROS de MLP:** O balanceamento de carga CEF é aplicável somente ao tráfego IP. O MLP equilibra todo o tráfego enviado por um pacote. O MLP mantém o pedido de pacotes. O próprio IP é tolerante a reordenações, portanto isso pode não importar para você; de fato, o custo adicional envolvido na manutenção do sequenciamento pode ser uma razão para evitar o MLP. O IP destina-se a redes que podem fornecer datagramas fora de ordem e qualquer coisa que use o IP deve ser capaz de lidar com a reorganização. No entanto, apesar disso, a realidade é que a reorganização ainda pode representar um problema real. O MLP fornece uma única conexão lógica com o sistema peer. A QoS é suportada em pacotes Multilink. O MLP fornece recursos dinâmicos de largura de banda, pois o usuário pode adicionar ou remover links de membros com base nas necessidades atuais. O MLP pode agrupar um número maior de links, enquanto o balanceamento de carga CEF é limitado a 6 caminhos de IP paralelos. O balanceamento de carga CEF por fluxo limita a largura de banda máxima de qualquer fluxo específico a um T1. Por exemplo, os clientes que usam gateways de voz podem ter muitas

chamadas com a mesma origem e destino e, portanto, usar apenas um caminho. **CONS de MLP:** O MLP adiciona uma sobrecarga extra a cada pacote ou quadro. O MLP é intensivo de CPU; dMLP é uma utilização intensiva da CPU da placa de linha.

10. **Como posso configurar vários pacotes entre dois roteadores?** Multilink determina qual pacote um link será associado com base no nome do peer e no discriminador de ponto de extremidade. Para criar vários pacotes distintos entre dois sistemas, o método padrão é forçar alguns dos links a se identificarem de forma diferente. O método recomendado é o uso do comando `ppp chap hostname name`.
11. **Posso ter links de membros de PAs diferentes?** Não. Se você deseja executar o dMLP, ele não é suportado. No entanto, se os links de membros forem adicionados de PAs diferentes, o controle será fornecido ao RSP e seu dMLP não será mais. O MLP ainda está funcionando, mas os benefícios do dMLP desapareceram.
12. **Posso misturar links de membros de ambos os compartimentos?** Não. Se você deseja executar o dMLP, ele não é suportado. No entanto, se forem adicionados links de membros de PAs diferentes, o controle será fornecido ao RSP e não será mais dMLP. O MLP ainda está funcionando, mas os benefícios do dMLP desapareceram.
13. **Posso ter links de membros entre VIPs ou FlexWANs diferentes?** Não. Se você deseja executar o dMLP, ele não é suportado. No entanto, se os links de membros forem adicionados de PAs diferentes, o controle será fornecido ao RSP e seu dMLP não será mais. O MLP ainda está funcionando, mas os benefícios do dMLP desapareceram.
14. **Posso ter links de membros entre portas diferentes de um único PA?** (Por exemplo, um link de membro de cada porta CT3 de um PA-MC-2T3+.) Yes. Desde que seja do mesmo PA, não há problemas.
15. **Posso agrupar portas T3 ou E3?** Não. Somente velocidades DS0, n\*DS0, T1 e E1 são permitidas com dMLP para 7500/VIP, 7600/FlexWAN e 7600/FlexWAN2. **Observação:** o MLPPP distribuído é suportado somente para links membros configurados em velocidades T1/E1 ou T1/E1 de subtaxa. As interfaces STM-1/T3/T1 canalizadas também suportam dMLPPP em velocidades T1/E1 ou T1/E1 de subtaxa. O MLPPP distribuído não é suportado para links membros configurados em T3/E3 de canal livre ou velocidades de interface mais altas.
16. **O que são fragmentos "reordenados"?** Se o fragmento ou pacote recebido não corresponder ao número de sequência esperado, o contador `reordenado` será incrementado. Para tamanhos de pacote variados, isso deve acontecer. Para pacotes de tamanho fixo, isso também pode acontecer porque o driver PA processa os pacotes recebidos em um enlace e não se baseia em rodízio (como é feito em dMLP durante a transmissão dos pacotes). Reordenado não significa perda de pacote.
17. **O que são fragmentos "perdidos"?** Sempre que o fragmento ou pacote é recebido fora de ordem e você descobre que fragmentos ou pacotes fora de ordem são recebidos em todos os links, o contador de `fragmentos perdidos` é incrementado. Outro caso é quando os fragmentos fora de ordem estão sendo armazenados na lista e atingem um limite (decidido com base na SRAM no VIP e o que for atribuído ao pacote), o contador de fragmentos perdidos é incrementado e o próximo número de sequência na lista é levado para processamento.
18. **Como o dMLP detecta fragmentos perdidos?** Números de sequência: Se você estiver esperando a chegada de um fragmento com o número de sequência N, e todos os links receberem um fragmento com um número de sequência superior a N, você sabe que o fragmento N deve ser perdido, porque não há como ele chegar legalmente por trás de fragmentos com numeração mais alta no mesmo link. tempo limite: Se você ficar sentado

esperando por um fragmento, você o declarará como perdido e seguirá em frente. Sobrecarga do buffer de remontagem: Se você estiver esperando a chegada do fragmento N, e enquanto outros fragmentos (com números de sequência superiores a N) estão chegando em alguns dos links, você terá que estacionar esses fragmentos em um buffer de remontagem até que o fragmento N apareça. Há um limite para quanto você pode armazenar em buffer. Se o buffer sobrecarregar, você declarará novamente o fragmento N como perdido e retomará o processamento com o que estiver no buffer.

19. **O que são "recebidas perdidas"?** Há duas razões possíveis para fragmentos ou pacotes recebidos perdidos: Se o fragmento ou pacote recebido estiver fora da janela de intervalo de sequência esperada, o pacote será descartado marcando-o como recebido perdido. Se o fragmento ou pacote recebido estiver dentro da janela de intervalo de sequência esperada, mas você não puder alocar um cabeçalho de pacote para reparar esse pacote, o pacote será descartado e marcado como recebido perdido.
20. **A criptografia é compatível com dMLP?** No.
21. **Oferecemos suporte à compactação de cabeçalho PFC?** Não, não no caminho distribuído. O roteador da extremidade oposta não é recomendado para configurar a compactação do cabeçalho PFC porque voltamos para o modo não distribuído se recebermos quadros ou pacotes de cabeçalho compactados. Se você quiser continuar executando dMLP, a compactação do cabeçalho PFC deve ser desabilitada em ambas as extremidades.
22. **A compactação de software é suportada com dMLP?** Não, porque a compactação de software não funcionará no caminho distribuído.
23. **A fragmentação é suportada no lado de transmissão?** Não com Vanilla dMLP. Não há problemas com o recebimento de fragmentos com Vanilla dMLP, mas no lado de transmissão a fragmentação não acontece. A fragmentação do lado de transmissão é suportada quando o **ppp multilink interleave** é configurado na interface dMLP.
24. **Podemos fazer ping nos links de membros de um pacote MLP?** Não, você não pode configurar um endereço IP nos links de membros.
25. **Existe alguma dependência no tamanho do fragmento MTU e MLP do link?** Não. O tamanho da MTU não tem nada a ver com o tamanho do fragmento MLP, além da restrição óbvia de que um fragmento MLP, como qualquer outro quadro, não pode exceder o tamanho da MTU dos links seriais.
26. **É possível configurar dois pacotes MLP entre um único par de roteadores?** Sim, é possível. No entanto, isso pode levar a uma diminuição do balanceamento de carga. Ele pode ser útil em testbeds, para simular mais de dois roteadores usando apenas dois roteadores, mas não tem nenhum valor real óbvio. Todos os links que vão para um peer comum *devem* ser colocados no mesmo pacote. Por definição, um pacote é o conjunto de links que vão para um peer específico. Um "peer" é identificado pelos valores de Username e Endpoint Discriminator que ele oferece durante as fases de LCP e autenticação. Se você estiver tentando criar vários pacotes entre dois roteadores, isso significa que você está tentando fazer com que cada máscara de roteador seja mais do que um único peer para sua contraparte. Devem identificar-se adequadamente.
27. **Os links de membros podem ter algoritmos de enfileiramento diferentes?** Todos os mecanismos de enfileiramento relacionados a um pacote precisam ser aplicados no nível do pacote e não no nível do link do membro. No entanto, a configuração de um algoritmo de fila não deve afetar como os pacotes são comutados para fora do pacote.
28. **Por que o tx-queue-limit é definido como 26 como padrão para links de membros para um pacote multilink quando o dMLP é ativado em um Cisco 7500?** Para qualquer interface serial de largura de banda T1/E1, o limite de fila tx é cerca de 4 ou 5. Quando você agrupa

T1s/E1s em multilink, a largura de banda aumentaria para o pacote. Como a comutação ocorreria com base na largura de banda da interface MLP, você precisa aumentar o tx-queue-limit dos links membros. Apenas um dos links membros, chamado de link primário, é usado para switching, portanto, seu limite de fila de tx precisa ser aumentado. Além disso, esse valor é um empírico escolhido após o teste e, em seguida, ajustado para esse valor. Em geral, as implantações não têm mais de 4 a 6 links T1/E1 em um pacote. Um valor de 26 pode atender perfeitamente a 6 a 8 links T1/E1 e, portanto, esse valor foi escolhido.

29. **O que é atraso diferencial e seu valor na implementação dMLP?** O dMLP suporta um atraso diferencial de 30 ms. Isso significaria que um fragmento é recebido por vez T e está fora de ordem (esperando um número de sequência 100, mas recebemos 101). Se o número de sequência 100 não for recebido até T+30 ms, 100 seria declarado perdido e se você puder iniciar o processamento a partir de 101, você faria isso. Caso não seja possível começar com 101 (se for um fragmento do meio), você deve procurar o fragmento que tem o fragmento inicial (por exemplo, 104) e começar a partir daí.
30. **O que acontece quando os pacotes são fragmentados no nível IP com multilink no 7500?** Se os pacotes forem fragmentados no nível IP, eles serão transportados sem remontagem nos saltos intermediários, mas remontados no roteador de destino.
31. **O que acontece quando os pacotes são fragmentados no nível MLP no 7500?** Se os pacotes forem fragmentados no nível MLP e se os pacotes reagrupados forem maiores que MRRU, os pacotes serão descartados em multilink. A fragmentação do lado da transmissão é suportada em dMLP somente com dLFI. Os pacotes são fragmentados no nível MLP somente se o tamanho do pacote for maior que o tamanho do frame e menor que o MRRU. Se forem enviados mais pacotes do que a MRRU e se ela não for fragmentada no nível IP, a outra extremidade descartará todo o tamanho dos pacotes que não são fragmentados no nível MLP porque os pacotes são mais do que a MRRU.
32. **Como se calcula o MRRU?** O MRRU é calculado de acordo com estas preferências: Para os novos links de membros que chegam, o MRRU é negociado novamente no nível de LCP de acordo com o MRRU configurado nos links de membros. O valor configurado na interface de link com o comando `ppp multilink mrru interface`. Se não estiver configurado, o valor herdado da configuração do comando `ppp multilink mrru` na interface pai. Se ambos os valores estiverem presentes, o valor da interface do link terá precedência. O MRRU padrão de 1524.

## [Aprimoramentos de depuração](#)

Esses aprimoramentos serão abordados no futuro. O planejamento ainda não foi concluído.

- Ative o comando `debug frame-relay multilink` no LC.
- Aprimore as CLIs de depuração atuais por interface e o número especificado de pacotes.
- Para dDDR, a funcionalidade de QoS ainda não é suportada. Isso só pode ser feito com um caso de negócios apropriado.

## [Informações Relacionadas](#)

- [CEF do discador](#)
- [Configuração de DDR peer-to-peer com perfis de discagem](#)
- [MPLS—Suporte PPP multilink](#)

- [Multilink Point-to-Point Protocol distribuído para Cisco 7500 Series Routers](#)
- [Frame Relay Multilink distribuído \(FRF.16\)](#)
- [Fragmentação e intercalação de link distribuído sobre linhas alugadas](#)
- [Links de VoIP por PPP com qualidade de serviço \(LLQ / prioridade IP RTP, LFI, cRTP\)](#)
- [Notas técnicas para solução de problemas - Cisco 7500 Series Router](#)
- [Página de suporte do produto do roteador - Cisco Systems](#)
- [Suporte Técnico e Documentação - Cisco Systems](#)