

Configuração e verificação de características distribuídas em Cisco 75xx e 76xx Router

Índice

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Convenções](#)

[Características distribuídas](#)

[MLPPP distribuído](#)

[LFI distribuído](#)

[Diferenças entre o dMLP e o dLFloLL](#)

[MLFR distribuído](#)

[DDR distribuído](#)

[Condições prévias e limitações distribuídas das características](#)

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

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

[Vida de um pacote](#)

[Trajeto de dados RX](#)

[Trajeto de dados de Tx](#)

[Remontagem](#)

[Configurando, verificando, e debugando características distribuídas](#)

[Configurando e verificando o dMFR](#)

[Configurando e verificando o dMLP/dLFloLL](#)

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

[Configurando e verificando o dDDR](#)

[Debugando o dMLP e o dDDR](#)

[Perguntas mais freqüentes](#)

[Debugar realces](#)

[Informações Relacionadas](#)

[Introdução](#)

Este documento ajuda-o a compreender, configurar, e verificar estas características:

- Protocolo ponto-a-ponto distribuído do Multilink (dMLP)
- Fragmentação de link distribuído e intercalação (LFI)
- LFI distribuído sobre a linha alugada (dLFloLL)
- LFI distribuído sobre o Frame Relay (dLFloFR)

- LFI distribuído sobre ATM (dLFloATM)
- Diferenças entre MLP distribuído (dMLP) e dLFloLL
- Frame Relay distribuído do Multilink (dMLFR)
- Roteamento por encomenda distribuído do seletor (DDR)

Pré-requisitos

Requisitos

Os leitores deste documento devem ter o conhecimento de características distribuídas para Cisco 7500/7600/6500.

Componentes Utilizados

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

- Todo o Cisco 7500 e 7600 Plataformas **Nota:** A informação neste documento igualmente aplica-se a 6500 Plataformas.
- Software release relevantes de Cisco IOS®, que esta tabela alista:

Apoio distribuído das características para cada ramo e plataforma

Recurso	Adaptadores de porta (PA) ¹ apoiado	7500 Plataformas		7600 Plataformas	
		Liberações de Cisco IOS Software principal	O Cisco IOS libera-se (o ínterim)	Liberações de Cisco IOS Software principal	Cisco IOS Software Release (ínterim)
dMLP	Sobrenome-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 e 12.0(9)S mais atrasado e mais tarde	12.2SX 12.1E ²	
dLFloLL	Sobrenome-PA PA-4T+ PA-8T	12.2T 12.3 12.3T 12.0S	12.2(8)T e 12.0(24) S mais atrasado e mais tarde	12.2SX	12.2(17)SXB e mais tarde
dLFloFR	Sobrenome-PA PA-4T+ PA-8T	12.2T 12.3 12.3T	12.2(4)T 3 e mais tarde	12.2SX	12.2(17)SXB e mais

					tarde
dLFlo ATM	PA-A3 PA-A6	12.2T 12.3 12.3T	12.2(4)T 3 e mais tarde	12.2SX	12.2(1 7)SXB e mais tarde
dMLF R	Sobreno me-PA PA-4T+ PA-8T	12.0S 12.3T	12.0(24) S e 12.3(4)T mais atrasado e mais tarde	12.2SX	12.2(1 7)SXB e mais tarde
QoS no dMLP	Sobreno me-PA PA-4T+ PA-8T	12.0S 12.2T 12.3 12.3T	12.0(24) S e 12.2(8)T mais atrasado e mais tarde	12.2SX	12.2(1 7)SXB e mais tarde
MPLS em dMLP MPLS no dLFlo LL	Sobreno me-PA PA-4T+ PA-8T	12.2T 12.3	12.2(15) T11 e mais tarde 12.3(5a) e mais atrasado	12.2SX	12.2(1 7)SXB e mais tarde
DDR distribuído	PA-MC- xT1 PA- MC-xE1 PA-MC- xTE1 PA- MCX- xTE1	12.3T	12.3(7)T e mais tarde		

Nota: Esteja ciente desta informação:

1. Estas características distribuídas apoio PA: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 apoia estas características em 7500 e 7600 Plataformas.

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a sua rede estiver ativa, certifique-se de que entende o impacto potencial de qualquer comando.

[Convenções](#)

Para obter mais informações sobre convenções de documento, consulte as [Convenções de dicas técnicas Cisco](#).

Características distribuídas

Estas características são explicadas neste documento:

- MLP distribuído
- LFI distribuído
- LFI distribuído sobre a linha alugada
- LFI distribuído sobre o Frame Relay
- LFI distribuído sobre o ATM
- Diferenças entre o dMLP e o dLFIoLL
- MLFR distribuído
- Discador distribuído
- Plataformas e placas de linha que apoiam características distribuídas

MLPPP distribuído

A característica distribuída do protocolo ponto-a-ponto do Multilink (dMLP) permite que você combine as linhas T1/E1 completas ou fracionárias em uma placa de linha (VIP, FlexWAN) em um Cisco 7500 ou 7600 Series Router em um pacote que tenha os links combinados da largura de banda de múltiplas. Você usa um conjunto MLP distribuído para fazer este. Um usuário pode escolher o número de pacotes em um roteador e o número de links pelo pacote. Isto permite que o usuário aumente a largura de banda dos links de rede além daquela de uma única linha T1/E1 sem a necessidade de comprar uma linha T3. No non-dMLP, todos os pacotes são comutados pelo route processor (RP); conseqüentemente, esta aplicação impacta o desempenho do RP, com a utilização elevada da CPU para somente algumas linhas T1/E1 que executam o MLP. Com dMLP, o número total de pacotes que podem ser segurados no roteador é aumentado, porque o trajeto de dados é asegurado e limitado pela placa de linha CPU e pela memória. o dMLP permite que você empacote o T1/E1 fracionário, partindo de DS0 (64 kbps) avante.

LFI distribuído

Os suportes de recurso dLFI o transporte do tráfego de tempo real (tal como a Voz) e do tráfego do tempo não real (tal como dados) no Frame Relay da velocidade mais baixa e nos circuitos virtuais ATM (VC) e em linhas alugadas sem causar o retardo excessivo ao tráfego de tempo real.

Esta característica é executada usando o Multilink PPP (MLP) sobre o Frame Relay, o ATM, e as linhas alugadas. A característica fragmenta um grande pacote de dados em uma sequência de fragmentos menores, para permitir pacotes sensíveis a retardo do tempo real e os pacotes do tempo não real para compartilhar do mesmos ligam. Os fragmentos são intercalados então com os pacotes do tempo real. No lado receptor do link, os fragmentos são remontados e o pacote é reconstruído.

A característica dLFI é frequentemente útil nas redes que enviam o tráfego de tempo real através do Distributed Low Latency Queuing (tal como a Voz) mas tem problemas de largura de banda. Isto atrasa o tráfego de tempo real devido ao transporte de grande, menos pacotes de dados sensíveis ao tempo. Você pode usar a característica dLFI nestas redes, para desmontar os grandes pacotes de dados em segmentos múltiplos. Os pacotes do tráfego de tempo real então podem ser enviados entre estes segmentos dos pacotes de dados. Nesta encenação, o tráfego de tempo real não experimenta um atraso longo quando esperar os pacotes de dados de prioridade baixa para atravessar a rede. Os pacotes de dados são remontados no lado receptor

do link, assim que os dados são intactos entregado.

O tamanho do fragmento do link é calculado com base no retardo de fragmentação no conjunto multilink, configurado com o **comando ppp multilink fragment-delay n**, onde:

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

Este tamanho do fragmento representa a virulência IP somente. Não inclui os bytes de encapsulamento (tamanho do fragmento = peso – bytes de encapsulamento). Os termos “peso” e “tamanho do fragmento” são como visto na saída do **comando show ppp multilink no RP**. Se o retardo de fragmentação não é configurado, o tamanho do fragmento do padrão está calculado para um retardo de fragmentação máximo de 30.

Nota: Com links da largura de banda de variação, o tamanho do fragmento escolhido é baseado no link com menos largura de banda.

[LFI distribuído sobre a linha alugada](#)

A característica dLFloLL estende a funcionalidade da fragmentação de link distribuído e da intercalação às linhas alugadas. O LFI distribuído é configurado com o **comando ppp multilink interleave na** relação do grupo multilink. É aconselhável que você use o LFI distribuído em interfaces multilink com a largura de banda menos de 768 kbps. Isto é porque o retardo de serialização para 1500 pacotes de bytes para maiores de 768 kbps da largura de banda está dentro dos limites do atraso aceitável e não precisa de ser fragmentado.

[LFI distribuído sobre o Frame Relay](#)

A característica dLFloFR é uma extensão do Multilink PPP sobre a característica do Frame Relay (MLPoFR). O MLP é usado para a fragmentação. Esta característica é similar ao FRF.12, que apoia a fragmentação e pode intercalar pacotes de alta prioridade através do Enfileiramento da latência baixa.

O **comando ppp multilink interleave no Virtual-molde** é exigido permitir a intercalação na interface de acesso virtual associada. Além do que a possibilidade do interruptor do CEF distribuído na interface serial, este comando é uma condição prévia para que o LFI distribuído trabalhe.

Nota: A menos que você estiver usando o Frame Relay ao inter-rede ATM, recomenda-se que você use o FRF.12 um pouco do que o dLFloFR, porque a utilização da largura de banda é melhor com FRF.12

[LFI distribuído sobre o ATM](#)

A característica dLFloATM é uma extensão do Multilink PPP sobre a característica ATM (MLPoATM). O MLP é usado para a fragmentação.

O **comando ppp multilink interleave no Virtual-molde** é exigido permitir a intercalação na interface de acesso virtual associada. Além do que a possibilidade do interruptor do CEF distribuído na interface serial, este comando é uma condição prévia para que o LFI distribuído trabalhe.

Com dLFloATM, é muito importante que você selecione um tamanho do fragmento que faça os pacotes para caber nas células ATM de tal maneira que não causam o estofamento desnecessário nas células ATM. Por exemplo, se o tamanho do fragmento selecionado é 124

bytes, isto significa que uma virulência IP de 124 bytes iria finalmente como $124 + 10$ (cabeçalho MLP) + 8 (cabeçalho SNAP) = 142 bytes. É importante notar que o primeiro fragmento sairia com $124 + 10 + 2$ (primeiros tamanho de cabeçalho do fragmento PID) + $8 = 144$ bytes. Isto significa que este pacote usará três células ATM para transferir o payload e, daqui, usaria a pilha embalada o mais eficientemente.

Diferenças entre o dMLP e o dLFloLL

o dMLP não apoia a fragmentação no lado de transmissão, visto que o dLFloLL faz.

Nota: A intercalação e a fragmentação usadas com mais de um link no conjunto multilink para o tráfego de voz não garantem que o tráfego de voz recebido através dos links múltiplos no pacote estará recebido em ordem. Pedir correto da Voz é assegurado nas camadas superior.

MLFR distribuído

A característica distribuída MLFR introduz a funcionalidade baseada no acordo da execução do Frame Relay UNI/NNI do Multilink do fórum do Frame Relay (FRF.16) aos Cisco 7500 e 7600 Series Router placa de linha-permitidos. A característica distribuída MLFR fornece uma maneira eficaz na redução de custos de aumentar a largura de banda para aplicativos particulares porque permite que os links de seriais múltiplos sejam agregados em um único pacote de largura de banda. O MLFR é apoiado nas User-to-Network Interfaces (UNI) e nas interfaces rede a rede (NNI) nas redes do Frame Relay.

O pacote é composto dos links de seriais múltiplos, chamado links de pacote. Cada link de pacote dentro de um pacote corresponde a uma interface física. Os links de pacote são invisíveis à camada de link de dados do Frame Relay, assim que a funcionalidade do Frame Relay não pode ser configurada nestas relações. A funcionalidade regular do Frame Relay que você quer aplicar a estes links deve ser configurada no bundle interface. Os links de pacote são visíveis aos dispositivos de peer.

DDR distribuído

A característica distribuída DDR permite o Distributed Switching em interfaces do discador. Sem esta característica, todo o tráfego de discado deve ser punted ao host para comutar. Com ele, somente os pacotes de controle são enviados até o RP, visto que a decisão de switching está feita nos VIP eles mesmos depois que uma conexão foi estabelecida.

A configuração do discador e a configuração do perfil de discador do legado são apoiadas somente com encapsulamento PPP. O MLP em interfaces do discador é apoiado igualmente. QoS não é apoiado com Distributed Switching em interfaces do discador.

Condições prévias e limitações distribuídas das características

Pré-requisitos

Estes são pré-requisitos gerais para todas estas características distribuídas:

- O Distributed Cisco Express Forwarding (dCEF) que comuta deve ser permitido globalmente.
- o switching dCEF deve ser permitido na interface serial do membro, que são parte de

conjunto MLP.

- o switching dCEF deve ser permitido no enlace físico de relações dLFloFR e dLFloATM.
- A configuração da intercalação é exigida para distribuir LFloFR e LFloATM.
- Configurar a largura de banda requerida na interface de molde virtual para relações dLFloFR e dLFloATM.
- Quando o PPP debuga estão permitidos no RP, você pôde observar o MLP: Encaminhado à mensagem da `interface errada` no Route Switch Processor (RSP). Porque esta mensagem é desconcertante e indesejável — especialmente se a mensagem é para pacotes do Cisco Discovery Protocol (CDP) — você deve configurar o **no cdp enable** nos enlaces membros do pacote.
- Todos os enlaces membros do pacote devem ter o `keepalive` permitido.

[Restrições](#)

Estas são restrições gerais para todas estas características distribuídas:

- As linhas de T1 and E1 não podem ser misturadas em um pacote.
- O retardo de diferencial apoiado máximo é a Senhora 30.
- Todas as linhas em um pacote devem residir no adaptador da mesma porta (PA).
- A compressão da ferragem não é apoiada.
- O VIP ou o FlexWAN CEF são limitados ao IP somente; todos protocolos restantes são enviados ao RSP.
- A fragmentação não é apoiada no lado de transmissão para o dMLP e o dMLFR.
- Muitos dos mecanismos de filas mais velhos não são apoiados pelo dLFI. Estes mecanismos incluem:Enfileiramento considerável em uma relação virtual do moldeAleatório-detecte em uma relação virtual do moldeEnfileiramento feito sob encomendaFilas de prioridade
- O enfileiramento considerável, a detecção aleatória (dWRED), e as filas de prioridade podem ser configurados em uma política de tráfego com o Modular QoS CLI.
- Somente um link pelo conjunto MLP é apoiado, quando você está usando o dLFloFR ou o dLFloATM. Se mais de um link é usado em um conjunto MLP ao usar o dLFloFR ou o dLFloATM, o dLFI está desabilitado automaticamente. Ao usar o dLFI sobre linhas alugadas, mais de um link pode ser configurado com o dLFI no conjunto MLP.
- Com dLFloATM, somente o `aal5snap` e `aal5mux` são apoiados. O `aal5nlpid` do encapsulamento e `aal5ciscopp` não são apoiados.
- Somente a Voz sobre o IP é apoiada; A Voz sobre o Frame Relay e o Voice over ATM não são apoiados pela característica 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 o LFI permitidoO conjunto multilink tem mais os enlaces membros de uma política de QoS com recursos de prioridade é permitida na interface multilink

Com dMLP e configuração dLFI, os pacotes de prioridade não levam o cabeçalho MLP e o número de sequência, e o MLP distribuirá os pacotes de prioridade através de todos os enlaces membros. Em consequência, os pacotes que são comprimidos pelo CRTP podem chegar fora de serviço no roteador de recepção. Isto proibe o CRTP de descomprimir o cabeçalho de pacote de informação e força o CRTP para deixar cair os pacotes.

[Recomendações](#)

Recomenda-se que os enlaces membros em um pacote têm a mesma largura de banda. Se você adiciona links de largura de banda desiguais ao pacote, conduzirá para lotear do pacote que requisita novamente, que fará com que a taxa de transferência total do pacote diminua.

O VIP2-50 (com 8 MB SRAM) ou é recomendado mais altamente ser usado com estas características distribuídas.

Número de pacotes e links e requisitos de memória

Refira o [protocolo multilink point-to-point distribuído para Cisco 7500 Series Router](#).

Hardware e software MLPPP ou MLFR em placas de linha de 7600 SORVOS

O MLP e o MLFR podem ser software ou com base em hardware. No MLP ou no MLFR baseado hardware, Freedm fornece a funcionalidade de multilink e os cabeçalhos MLP são adicionados pelo chip freedm. No MLP ou no MLFR baseado software, a placa de linha CPU do SORVO fornece a funcionalidade de multilink (que é similar às aplicações VIP e de FlexWAN).

Estas são as limitações e as circunstâncias para executar o MLP ou o MLFR baseado hardware.

- Pode haver somente um máximo de 336 pacotes pela placa de linha e de 168 pacotes pela avaliação da postura de segurança (TERMAS) (Freedm).
- Pode haver somente um máximo de 12 DS1/E1 pelo pacote.
- Todos os links devem pertencer aos mesmos TERMAS (Freedm).
- Todos os links no pacote devem se operar na mesma velocidade.
- O tamanho do fragmento TX pode ser 128, 256, ou 512. Um tamanho do fragmento configurado CLI é traçado ao tamanho do fragmento apoiado o 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 1 a 9.6 KB.
- O formato proprietário de Cisco não pode ser apoiado (MLFR).

No hardware LFI, se há somente um link no pacote e se aquele é DS1/E1 então a fragmentação e intercalação será feita por Freedm.

A saída do **multilink de PPP da mostra** mostra se a implementação de hardware está sendo executado.

```
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 o multilink é com base no software então a saída do **multilink de PPP da mostra** não terá o Multilink no hardware na saída.

Vida de um pacote

Trajeto de dados RX

1. Pacote recebido pelo direcionador.
2. O encapsulamento é verificado: como segue Encapsulamento básico: No dMLP, o tipo de encapsulamento para a interface de ingresso é ET_PPP. No dMLFR, o tipo de encapsulamento para a interface de ingresso é ET_FRAME_RELAY. No dLFloLL, o tipo de encapsulamento para a interface de ingresso é ET_PPP. No dLFloFR, o tipo de encapsulamento para a interface de ingresso é ET_FRAME_RELAY. No dLFloATM, o tipo de encapsulamento para a interface de ingresso é ET_ATM. No dDialer, o tipo de encapsulamento é ET_PPP. Processamento adicional do encapsulamento: Para o ET_PPP, o NLPID é aspirado para fora. Para o dMLP, o NLPID é MULTILINK. Para o dLFloLL, há duas coisas a considerar: Pacotes voip — Estes não têm um cabeçalho MLP e têm um NLPID que indique o IP. Pacotes de dados — O NLPID é MULTILINK. Para o dDialer, os pacotes não terão um cabeçalho MLP e têm um NLPID que indicam o IP. **Nota:** Neste caso, você pode configurar dCRTP (protocolo compressed real-time distribuído). Em caso afirmativo, o encabeçamento é descomprimido antes do processamento adicional.
3. Para o ET_FRAME_RELAY, se o link em que o pacote está recebido é configurado para o dMLFR então o pacote é processado para o dMLFR
4. Para o dLFloFR e o 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 o dCRTP é configurado, o encabeçamento está descomprimido antes do processamento adicional. Os pacotes de voz são comutados imediatamente. Um pacote de dados fragmentados terá que ser remontado antes que esteja comutado. Com ET_PPP, você pôde vir através dos pacotes do link de PPP; e com ET_FRAME_RELAY, você pôde vir através dos pacotes de controle MLFR. Todos estes pacotes de controle punted ao RP para processar.
5. Segundo a descodificação acima mencionada, o pacote é verificado para ver se há o tipo de comutá-lo exige. O tipo de link determinará se o pacote deve IP-ser comutada ou MPLS-comutada. Os pacotes são dados então às funções de switching respectivas.
6. Com empacotamento conjuntamente com características distribuídas, o turbocompressor IP que comuta rapidamente o vetor é roubado. Isto é feito porque o pacote é recebido no enlace membro; contudo, deve-se tratar tais que está recebido no pacote. Você igualmente precisa de verificar para ver se há pacotes de controle que punted ao host. Principalmente no dMLFR, há os pacotes da interface de gerenciamento local (LMI) que não são pacotes de controle MLFR. Para estes, um diferente o espaço do número do dLCI é usado parte de. Sempre que o dLCI é descodificado para cair neste espaço, o pacote punted até o host, porque se reconhece para ser um pacote de LMI. Os pacotes voip (enfileirados na fila de latência baixa) são comutados apenas para fora sem a adição do cabeçalho MLP. As características distribuídas podem receber e remontar pacotes, quando os pacotes de dados fragmentados são recebidos. O processo de remontagem é explicado em uma seção mais recente. Se o pacote deve etiqueta-ser comutado, está passado então à rotina de tag-switching, no dMLP. Se não, se deve IP-ser comutada, é passado à rotina da Comutação IP. **Nota:** Todos os pacotes não-IP punted para hospedar, no dMLFR.
7. **IP:** A função da Comutação IP é comum a todos os pacotes. Faz principalmente três coisas: Faça o processamento necessário dos pacotes, caso que todas as características

são configuradas. Igualmente, quando o discador distribuído é usado, faça atualizações do temporizador de ociosidade aqui quando um “pacote interessante” é recebido. Refira o quietude-[intervalo do discador \(relação\)](#), o [dialer fast-idle \(relação\)](#), e [configurar um perfil do discador](#) para detalhes do parâmetro de configuração do **temporizador de ociosidade**. Em 75xx Router, a adjacência indicará o `tx_acc_ptr` para a interface de saída. Se a interface de saída é uma interface de acesso virtual, o `tx_acc_ptr` é NULO. Neste caso, fixe acima o encapsulamento e obtenha o `tx_acc_ptr` do `hwidb` `mentir`. Estes consulta e encapsulamento fixam são acima necessários no dLFloFR e no dLFloATM. No dLFloLL, o link é tratado como parte de um conjunto multilink. **Nota:** O TTL para o pacote é ajustado aqui, e a verificação para a fragmentação de IP é feita. O `mci_status` é ajustado a `RXTYPE_DODIP` para todos os pacotes.

8. Com a decisão de switching feita, o pacote está pronto para ser enviado para fora da relação. A relação está verificada para determinar se apoia o switching local. Se faz, está mandada diretamente através do `fastsend`. Se não, uma tentativa é feita ao interruptor do cache de rota ele. Note que, caso que QoS é configurado para a relação, o vetor do switching local está roubado por QoS. O HQF enviará à fila o pacote e processará mais o pacote, antes que esteja enviado finalmente fora da relação. Este é o caso com dLFI. Para o dLFI, a fragmentação e a intercalação são ajustadas. QoS segura a invocação de nossa rotina de fragmentação e intercala os pacotes fragmentados com os pacotes de voz que estarão enfileirados na fila de prioridade (se o LLQ é configurado). Isto assegura-se de que os pacotes voip não sofram do atraso exigido para enviar pacotes de dados enormes através do link.

[Trajeto de dados de Tx](#)

O `vip_dtq_consumer` obtém o pacote e obtém o número de interface, de que obtém o `BID`. A rotina de envio rápido que corresponde ao `BID` é chamada:

i) Fastsend

1. No dMFR, a estrutura do `fr_info` é recuperada da tabela que combina o `if_index` ao `fr_info`. Os pacotes de controle são mandados apenas. O cabeçalho de frame dará o dLCI, que o ajudará a determinar se este é um pacote de LMI ou um pacote de dados. O campo do `dlci` no cabeçalho de frame overwritten com o número de sequência do `dmfr`. Os números de sequência separados são usados para o LMI e os pacotes de dados. **Nota:** Os números de sequência separados são usados para dLCIs separados.
2. No dMLP, os pacotes de controle são enviados com conjunto de prioridades à elevação. Com pacotes de dados, se o dCRTP é configurado, o encabeçamento é comprimido. O cabeçalho MLP VIP que inclui a ordenação de informação é adicionado e enviado fora dos enlaces membros.
3. No dLFI, o HQF intercepta os pacotes a ser enviados através da relação. Se é um pacote de voz, o pacote de voz está colocado na fila de prioridade (se o LLQ é configurado) e enviado fora da relação sem o encapsulamento MLP. Com pacotes de dados, chama o código da fragmentação dLFI, que retorna os fragmentos ao código de QoS, que são intercalados então com o tráfego de prioridade de modo que os requisitos de retardo do tráfego de voz sejam cumpridos. Também, se o dCRTP é configurado, simplesmente o encabeçamento para o pacote de voz é comprimido. Os encabeçamentos de pacote de dados são deixados enquanto são.

4. No dDialer, o pacote está classificado a fim restaurar o temporizador de ociosidade do link da saída antes que o pacote esteja mandado. Isto está feito depois que o link da saída está escolhido, caso diversos links estiverem limitados ao mesmo discador. Nenhum encabeçamento é adicionado aos pacotes de discadores. Assim, arranjando em sequência e remonte dos pacotes não são apoiados em interfaces do discador.

Nota: No dMLP, no dDialer, no dMLFR, e no dLFI com diversos links, o enlace físico em que o tráfego é enviado depende da congestão do link. Se o link é congestionado, mova-se para o link seguinte e assim por diante. (o dMLFR, o dMLP sem QoS, e as características do dDialer igualmente escolhem os links baseados no número de bytes que está sendo posto sobre o link. Escolhe o link seguinte, se o link atual tem transmitido já sua cota de bytes, em uma base round-robin. Esta quota é decidida pelos frag_bytes para o link. Para interfaces membro do discador, os frag_bytes são ajustados ao valor padrão da largura de banda de interface.)

Nota: Nas configurações HQF nas relações da saída VIP, o HQF rouba o vetor do dtq_consumer. O pacote DMA'd à saída VIP atravessa primeiramente a verificação HQF. Se QoS é configurado na interface de saída, o HQF retrocede dentro para processar o pacote, antes que o pacote esteja fastsent fora da relação.

Remontagem

As relações lisas do dDialer não apoiam a remontagem e arranjar em sequência. Para permitir isto em interfaces do discador, o MLP sobre interfaces do discador terá que ser configurado. Se isto é feito, o trajeto RX e de Tx é idêntico aos trajetos dMLP. Quando os pacotes são recebidos, o número de sequência está verificado contra o número de sequência esperado.

- Se os números de sequência combinam: Se o pacote é um pacote não fragmentado então a remontagem não está exigida. Continue com etapas mais adicionais do interruptor. Se o pacote é um fragmento, a seguir verifique o começo e termine bit e construa o pacote à medida que os fragmentos são recebidos.
- Se os números de sequência não combinam: Se o número de sequência está dentro do indicador previsto dos números de sequência a seguir o põe no classificado "unassigned fragmenta a lista." Mais tarde, quando um número de sequência esperado não é recebido, esta lista é verificada, caso que o pacote foi armazenado aqui. Se o número de sequência não está dentro do indicador, rejeite-o e relate-o "o fragmento perdido recebido." Se um intervalo ocorre mais atrasado ao esperar este pacote, o receptor resynced, e ele começa outra vez com o próximo pacote recebido.

Em todos aqueles casos, uma corrente de pacote de informação corretamente pedida é enviada fora desta relação. Se os fragmentos são recebidos, um pacote completo está formado e mandado então.

Configurando, verificando, e debugando características distribuídas

Esta seção cobre os **comandos show and debug** que estão disponíveis verificar e debugar cada um das características distribuídas.

Configurando e verificando o dMFR

Configuração de exemplo MFR

```
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.
```

Nota: A relação MFR é como uma outra interface FR e daqui apoia a maioria da configuração FR.

```
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.
```

Verifique o status de conjunto de 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
```

Isto indica que duas relações estão adicionadas corretamente, e uma relação não negociou ainda os mensagens MLFR LIP.

Para obter mais informação no pacote e em enlaces membros MFR, 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 Debug MFR](#)

Estes debugam são úteis pesquisar defeitos as edições aonde um link não obtém adicionado ao pacote.

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

Nota: Quando uma relação específica ou a interface serial MFR não são especificadas, esta permite debuga para todos os links MFR. Isto pode ser opressivamente, se o roteador tem um grande número links MFR.

Para debugar os pacotes MFR que são recebidos no RP, assim como debugar as atividades do controle MFR, isto debuga é útil:

```
debug frame-relay multilink
```

Nota: Sob o tráfego pesado, isto pode oprimir o CPU.

[Verifique o status de conjunto de dMLFR no LC](#)

```
multilink do show frame-relay
```

Nota: Atualmente, isto não está disponível no LC, mas será adicionado logo. Até lá, **multilink de PPP da mostra do uso.**

```
debug frame-relay multilink
```

[Configurando e verificando o dMLP/dLFloLL](#)

[Configuração do Multilink PPP](#)

```
debug frame-relay multilink
```

Configuração de exemplo sob a interface serial:

```
debug frame-relay multilink
```

Nota: O comando `ppp chap hostname M1` não significa realmente que a autenticação chap está permitida. A corda **M1** neste comando atua como o fim-ponto-discriminador e é exigida somente se está indo estar mais de um conjunto multilink entre os mesmo dois Roteadores. Em tal caso, todos os links que pertencem a um pacote devem não ter o mesmo discriminador de ponto final, e nenhum dois links que pertence a um pacote diferente devem ter o mesmo discriminador de ponto final.

[Parâmetros da configuração opcional](#)

intercalação do multilink de PPP do [no]

Isto permite a intercalação no conjunto multilink. Isto trabalha conjuntamente com o Modular QoS CLI. Os pacotes de alta prioridade serão transmitidos sem a adição da sequência e de encabeçamento MLP, quando outros pacotes serão fragmentados e transmitidos com a sequência e o encabeçamento MLP.

Nota: Quando intercalar é permitida com mais de um link, é possível que o tráfego de alta prioridade obterá requisitado novamente. Quando intercalar é permitida ou desabilitada, uma restauração do pacote está exigida para obtê-lo ativado na placa de linha.

```
ppp multilink mrru local value
```

Isto especifica o Maximum Receive Unit no multilink; os pacotes até este tamanho serão aceitados pela interface multilink. O tamanho aqui exclui o cabeçalho MLP.

```
ppp multilink mrru remote value
```

Isto especifica o mínimo MRRU que a extremidade remota deve apoiar. Se a extremidade remota MRRU é menos do que este valor, a seguir a negociação do pacote falhará.

```
ppp multilink fragment delay seconds
```

Isto especifica o atraso permitido nos milissegundos (Senhora) causados por um fragmento de dados. Ou seja o valor do atraso é usado para computar o tamanho do fragmento máximo permitido. A aplicação distribuída difere da aplicação do Cisco IOS nestas maneiras:

1. A fragmentação não é executada a menos que intercalar for permitida.
2. Com links da largura de banda de variação, o tamanho do fragmento escolhido é baseado em menos relação da largura de banda.

```
ppp multilink fragment disable
```

Este comando não adiciona nenhuma funcionalidade na aplicação distribuída. A fragmentação ocorre somente quando intercalar é permitida; e, quando intercalar é permitida, o comando `ppp multilink fragment disable` é ignorado.

[Verifique o status de conjunto 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 reage do modo distribuído, este está indicado na saída do **multilink de PPP da mostra**:O pacote é distribuído. Se não, o pacote por qualquer motivo não é distribuído então.
2. Quando a **intercalação do multilink de PPP** é configurada e permitida na placa de linha, a saída do **multilink de PPP da mostra** inclui as estatísticas de dLFI onde:
Fragmentado — Indica a contagem dos fragmentos que foram transmitidos e recebidos.
Não-fragmentado — Indica a contagem dos pacotes que foram transmitidos ou recebidos sem obter fragmentados.
Remontado — Indica o número de pacotes completos que foram remontados.

Quando intercalar não é permitida, a saída olha como 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 de conjunto 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 dMFR, os números de sequência são mantidos em uma base por-DLCI, com o número de seqüência no pacote usado para o dLCI LMI.

Campo	Descrição
dmlp_ipc_config_count	Número de mensagens IPC recebidos pelo LC para o multilink ou a configuração MLFR
dmlp_bundle_count	O número de MLP e de MLFR empacota no LC
dmlp_ipc_fault_count	Número de mensagens de configuração que conduziram à falha no LC. Devem ser 0; se é diferente de zero então pôde haver um problema.
vetores da etiqueta	Indica o BID aos tag_optimum_fs e o BID aos vetores ip2tag_optimum_fs usados no switching de caractere.
board_encap	Indica o vetor do board_encap que está usado para adicionar 2 bytes do encapsulamento da placa, se há uns links separados em uma plataforma 7500. Deve ser NULO, se o link contém relações não-canalizados.
max_particles	Número máximo de partículas

	que podem ser realizadas no buffer de remontagem
mrru	O tamanho máximo de pacote que é aceitado sem considerar o encapsulamento MLP. Não aplicável para a relação MLFR.
seq_window_size	O tamanho máximo de janela para números de sequência
working_pak	Indica o pak atual sob a remontagem. ZERO, se nenhum.
working_pak_cache	Ponteiro à estática pak que é usada para a remontagem. Isto é atribuído quando o primeiro pacote NON-completo é recebido pelo pacote.
una_frag_list	Primeira entrada na fila da remontagem. Se a entrada não é NULA e não muda, indica que o temporizador não está executando uma questão de software.
una_frag_end	Última entrada na fila da remontagem
rcved_end_bit	Indica que o pacote recebeu um bit da extremidade, assim que ele está caçando para um bit do começo.
is_lost_frag	É verdadeiro, se um fragmento é declarado perdido. Isto obtém cancelado quando um fragmento com sequência esperada é recebido.
resync_count	Indica o número de vezes que o receptor era fora da sincronização com transmissor e tinha que resync começando com o último fragmento arranjado em sequência recebido.
intervalo	Indica que o timeout de remontagem ocorreu e pacotes estão sendo processados da fila da remontagem.
timer_start	O temporizador de remontagem do número de vezes foi ligado
timer_running	Indica mesmo se o temporizador de remontagem está sendo executado.
timer_count	Indica o número de vezes que o temporizador de remontagem

	expirou.
next_xmit_link	O link em que o próximo pacote será transmitido
Membro	Campo de bit que indica os membros atuais.
Congestão	Campo não usado em todos os ramos. Indica que enlaces membros não são congestionados.
dmlp_orig_pak_to_host	O vetor usou-se aos pacotes do pontapé ao RP.
dmlp_orig_fastsend	O envio rápido de driver original antes do MLP ou do MLFR alterou o fastsend do direcionador.
fragmentos perdidos	O número de fragmentos que foram perdidos (o receptor não recebeu estes fragmentos). Isto é cancelado periodicamente quando uma atualização é enviada ao host.
Requisitado novamente	Número de fragmentos que foram recebidos fora da ordem prevista. Isto é cancelado periodicamente quando uma atualização é enviada ao host.
Rejeitado	Número de fragmentos rejeitados porque um pacote completo não poderia ser feito
perdido recebido	O número de fragmentos recebeu que provavelmente foram perdidos. Isto indica que o atraso da ligação é maior do que o timeout de remontagem de dMLP da Senhora 30.

[Configurando e verificando o dLFloFR e o 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-templatel
!--- 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 o estado 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

```

Nota: O pacote tornar-se-á distribuído somente quando a intercalação do multilink de PPP é configurada sob o molde virtual; sem este comando, o pacote não será distribuído.

[Verifique o estado do pacote dLFIoFR/ATM no LC](#)

Para verificar o dLFI está trabalhando certamente corretamente no LC, emite 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 encap: 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 da prioridade e uma camada WFQ. A fragmentação será feita no bit de camada de intercalação de WFQ.

[Configurando e verificando o dDDR](#)

O DDR distribuído é ativado quando você permite o **cef IP distribuído** na configuração global e no **cache de rota IP distribuídos** nas interfaces do discador.

```
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 encap: 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
```

Não há nenhuma outra configuração especial para o DDR distribuído. Mais a configuração segue a configuração DDR normal.

[Verifique o roteamento por encomenda distribuído do seletor](#)

```
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
```

O tipo do discador diz-nos o tipo de discador usado. O ISDN implica a configuração do discador do legado e o PERFIL implica a configuração do perfil de discador. O estado do discador indica o estado atual do discador. O estado de uma interface do discador desligado será inativo. O temporizador de ociosidade é restaurado sempre que o tráfego interessante é considerado. Se este temporizador expira nunca, a relação desligará imediatamente. O temporizador de ociosidade é um parâmetro configurável. Para mais informações, refira [configurar o DDR peer-to-peer com Perfis de discagem](#).

```
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 que aquelas para o dMLP.

Debugando o dMLP e o dDDR

Debuga disponível no RP

dDDR

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

Emita este comando debugar e assim por diante funções de trajeto do controle como a configuração de chamada. Para mais informações, refira [eventos do debug dialer](#).

```
debug ip cef dialer
```

Emita este comando debugar eventos CEF-relacionados do discador. Para mais informações, refira o [Dialer CEF](#).

Debuga disponível no LC

dMLP

Eliminação de erros do trajeto do controle: **debugar o evento multilink**

Eliminação de erros do trajeto de dados: **debugar fragmentos do multilink**

Eliminação de erros do erro do trajeto de dados e do trajeto do controle: **debugar o erro de multilink**

Debugando o dMLP em placas de linha do SORVO

Despejando os pacotes baseados no CI: Os pacotes de dados e os pacotes de controle podem ser despejados nas placas de linha baseadas no ci do controle e no ci da sequência.

teste ci *CI-NUM* da descarga do *subslot_num* do módulo HW [RX|*num_packets_to_dump* TX]

Os CI podem ser obtidos desse modo:

```
!--- 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 Serial16/0/0:0 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_mdcs=403A9EEC
```

Para o CT3, você deve obter o `vc` numérico, que podem ser obtidos da saída da **relação CT3_interface_name de série da mostra**.

Agora a informação CI pode ser obtida do console dos TERMAS. Reoriente primeiramente a saída de comandos console dos TERMAS ao RP com o **comando `spa_redirect rp ct3_freedm336`**.

O comando `spa_ct3_test freedm show linkrec vc` mostra a informação necessária CI.

dMFR

Eliminação de erros do trajeto do controle: **debugar o evento do dmfr**

Eliminação de erros do trajeto de dados: **debugar pacotes do dmfr**

Eliminação de erros do erro do trajeto de dados e do trajeto do controle: **debugar o erro do dmfr**

Despejando os pacotes baseados no CI: Veja o [dMLP](#).

dLFI

Eliminação de erros do trajeto do controle: **debugar o evento do dlfi**

Eliminação de erros do trajeto de dados: **debugar fragmentos do dlfi**

Eliminação de erros do erro do trajeto de dados e do trajeto do controle: **debugar o erro do dlfi**

dDDR

Não há nenhum comando debugging especial; você deve usar o [dMLP debuga](#).

Em caso do dLFIoLL, o dMLP e o dLFI debugam puderam ter que ser usado. Estes debugam não são condicionais e, daqui, provocarão para todos os pacotes.

Perguntas mais freqüentes

1. **Que é dMLP?**o dMLP é curto para o Multilink PPP distribuído (como exposto no [RFC1990](#)). [Esta característica é apoiada por plataformas distribuídas, como o Cisco 7500 Series e o 7600 Series. o dMLP permite que você combine as linhas T1/E1 — em um VIP em um Cisco 7500 Series Router ou em um FlexWAN em um 7600 Series Router — em um pacote que tenha as linhas combinadas da largura de banda de múltiplas T1/E1. Isto permite que os clientes aumentem a largura de banda além do T1/E1 sem a necessidade de comprar uma linha T3/E3.](#)
2. **Que “é distribuído” no dMLP?**O termo “distribuído” implica que o packet switching está feito pelo VIP e não pelo RSP. Por quê? As potencialidades de switching RSP são um pouco

limitadas, e tem muito mais trabalhos importantes fazer. O VIP que é capaz dos pacotes de switching offloads esta atividade do RSP. O Cisco IOS com base em RSP ainda controla os links. A criação de pacote e o teardown são feitos pelo RSP. Adicionalmente, o processamento plano do controle PPP é feito ainda pelo RSP, incluindo a manipulação de todos os pacotes de controle PPP (LCP, autenticação, e os NCP). Contudo, uma vez que um pacote é estabelecido, a manipulação dos pacotes de MLP é virada ao VIP para comutar pelo CPU a bordo. O Engine de dMLP (no VIP) segura todos os procedimentos de MLP, incluindo a fragmentação, a intercalação, o encapsulamento, o Balanceamento de carga entre links múltiplos, e a classificação e a remontagem de fragmentos de entrada. As funções feitas pelo VIP em um sistema 7500 são feitas pelo FlexWAN/FlexWAN em um sistema baseado 7600.

3. Como eu sei se o pacote é distribuído ou não? Emita o comando show ppp multilink no console de 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)
    Sel1/1/0/27:0, since 00:22:46, no frags rcvd
    Sel1/1/0/25:0, since 00:22:46, no frags rcvd
```

!--- Output suppressed.

- 4. Se eu promovo ao RSP16 ou ao SUP720, meu desempenho de dMLP será melhor?** Não. O desempenho de switching do dMLP (ou de alguma característica distribuída) é dependente do VIP ou do FlexWAN na pergunta. Por exemplo, o desempenho de um VIP6-80 será melhor do que o desempenho com VIP2-50.
- 5. Que PA posso eu usar com esta característica?** 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 (7500 somente)
- 6. Quantos links podem ser configurados em um único pacote?** Há muitas facetas a esta resposta. O gargalo preliminar é a potência de CPU da placa de linha (VIP/FlexWAN/Enhanced-FlexWAN2). O limite duro é 56 links pelo pacote, mas muitas vezes que você não pode configurar aqueles muitos (e para ter que muito interruptor do tráfego), devido à potência de CPU ou aos buffers limitados. Estes números são baseados nesta diretriz (baseada no CPU e na memória no VIP/FlexWAN/Enhanced-FlexWAN2):(Com 4MB SRAM) T1s VIP2-50 máximo = 12(Com 8MB SRAM) T1s VIP2-50 máximo = 16T1s VIP4-80 máximo = 40T1s VIP6-80 máximo = 40O T1s máximo do FlexWAN = será atualizado logoAumentar-FlexWAN máximo E1 = 21 E1 pela baía (agregado 42 E1 pela placa de linha)
- 7. Há uma mudança no desempenho se eu configuro 3 pacotes com 3 T1s cada um ou 1 pacote com o T1s 9?** Não há nenhuma mudança no desempenho, como provado em testes de laboratório. Contudo, com um grande número T1s em um único pacote (diga 24 ou 28 T1s em um único pacote), há umas edições com ser executado fora dos buffers. Seu altamente recomendado que você para não ter mais de 8 enlaces 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. Seu a largura de banda agregada de todos os enlaces membros. Se você tem 4 T1s no pacote, a seguir a largura de banda do pacote é 6.144Mbps.

9. **Qual é melhor? equilíbrio da CEF-carga ou dMLP?** Não há nenhuma resposta simples a este. Suas necessidades decidem qual é melhor. **PROFISSIONAIS do MLP:** O Balanceamento de carga CEF é aplicável somente ao tráfego IP. O MLP equilibra todo o tráfego enviado sobre um pacote. O MLP mantém pedir dos pacotes. O IP próprio é tolerante da requisição, assim que este não pode importar-lhe; de fato, os custos extras envolvidos em manter arranjar em sequência podem ser uma razão evitar o MLP. O IP é pretendido para as redes que podem entregar as datagramas foras de serviço, e qualquer coisa que usa o IP é suposta para poder tratar a requisição. Contudo, apesar deste fato, a realidade é que requisitar novamente pode ainda levantar um problema real. O MLP fornece uma única conexão lógica ao sistema do par. QoS é apoiado em conjuntos multilink. O MLP fornece capacidades de largura de banda dinâmicas, como o usuário pode adicionar ou remover os enlaces membros baseados em necessidades atuais. O MLP pode empacotar números maiores de links, visto que o Balanceamento de carga CEF é limitado aos caminhos IP da paralela 6. o Balanceamento de carga do Por-fluxo CEF limita a largura de banda máxima de qualquer fluxo dado a um T1. Por exemplo, os clientes que usam o Gateways de voz podem mandar muitos atendimentos com a mesmos fonte e destino e, daqui, usar somente um trajeto. **CONS do MLP:** O MLP adiciona despesas gerais extra a cada pacote ou quadro O MLP é utilização de CPU; o dMLP é utilização de CPU da placa de linha.
10. **Como posso eu configurar pacotes múltiplos entre dois Roteadores?** O Multilink determina que pacote um link se juntará baseado no nome e no discriminador de ponto final do par. Para criar pacotes distintos múltiplos entre dois sistemas, o método padrão é forçar alguns dos links para identificar-se diferentemente. O método recomendada é o uso do **comando ppp chap hostname name**.
11. **Posso eu ter enlaces membros dos PA diferentes?** Não. Se você quer executar o dMLP, a seguir não está apoiado. Contudo, se os enlaces membros são adicionados dos PA diferentes, a seguir o controle é dado ao RSP e ao seu não dMLP anymore. O MLP ainda está funcionando, mas os benefícios do dMLP são idos.
12. **Posso eu misturar enlaces membros de ambas as baías?** Não. Se você quer executar o dMLP, a seguir não está apoiado. Contudo, se os enlaces membros são adicionados dos PA diferentes, a seguir o controle é dado ao RSP e não é dMLP anymore. O MLP ainda está funcionando, mas os benefícios do dMLP são idos.
13. **Posso eu ter enlaces membros através dos VIP ou de FlexWANs diferente?** Não. Se você quer executar o dMLP, a seguir não está apoiado. Contudo, se os enlaces membros são adicionados dos PA diferentes, a seguir o controle é dado ao RSP e ao seu não dMLP anymore. O MLP ainda está funcionando, mas os benefícios do dMLP são idos.
14. **Posso eu ter enlaces membros através das portas diferentes de um único PA?** (Por exemplo, um enlace membro de cada porta CT3 de um PA-MC-2T3+.) Sim. Enquanto é do mesmo PA, não há nenhuma edição.
15. **Posso eu empacotar as portas T3 ou E3?** Não. Somente as velocidades DS0, n*DS0, T1, e E1 são reservadas com o dMLP para 7500/VIP, 7600/FlexWAN, e 7600/FlexWAN2. **Nota:** O MLPPP distribuído é apoiado somente para os enlaces membros configurados em velocidades T1/E1 ou de subtaxa T1/E1. As relações STM-1/T3/T1 separadas igualmente apoiam o dMLPPP em velocidades T1/E1 ou de subtaxa T1/E1. O MLPPP distribuído não é apoiado para os enlaces membros configurados no canal desobstruído T3/E3 ou em umas velocidades mais altas da relação.
16. **Que são fragmentos "requisitados novamente"?** Se o fragmento ou o pacote recebido não combinam o número de sequência esperado, a seguir o contador `requisitado novamente` está incrementado. Para tamanhos do pacote de variação, isto é limitado para acontecer.

Para pacotes do tamanho fixo, isto pode igualmente acontecer porque o direcionador PA processa os pacotes que receberam em um link e não vão na base round-robin (como é feito no dMLP ao transmitir os pacotes). Reordered não significa a perda de pacotes.

17. **Que são fragmentos “perdidos”?** Sempre que o fragmento ou o pacote são fora de serviço recebido e você encontra que os fragmentos ou os pacotes foras de serviço estão recebidos em todos os links, os `fragmentos perdidos` são incrementados contra. Um outro caso é quando os fragmentos foras de serviço estão sendo armazenados na lista e alcança um limite (decidido com base no SRAM no VIP e o que quer que é atribuído para o pacote), os fragmentos perdidos é incrementado contra e o número de sequência seguinte na lista é tomado processando.
18. **Como o dMLP detecta fragmentos perdidos?** Números de sequência: Se você está esperando um fragmento com número de sequência N para chegar, e todos os links recebem um fragmento com um número de sequência mais alto do que N, você sabe que o fragmento N deve ser perdido, porque não há nenhuma maneira poderia legalmente chegar atrás de uns fragmentos numerados mais altos no mesmo link. Intervalo: Se você se senta demasiado por muito tempo esperando um fragmento, você eventualmente declará-lo-á como perdido e mover-se-á sobre. Excesso do buffer de remontagem: Se você está esperando o fragmento N para chegar, e entrementes os outros fragmentos (com os números de sequência mais altos do que N) estão chegando em alguns dos links, a seguir você tem que estacionar aqueles fragmentos em um buffer de remontagem até o fragmento N aparece. Há um limite a quanto você pode proteger. Se os excessos de buffer, você declaram outra vez o fragmento N como perdido, e o recomeçam processar com o que quer que está no buffer.
19. **O que são “perdidos recebidos”?** Há duas razões possíveis para fragmentos ou pacotes recebidos perdidos: Se o fragmento ou o pacote recebido são fora do indicador da escala da sequência esperada, o pacote está deixado cair marcando o enquanto perdido recebido. Se o fragmento ou o pacote recebido estão dentro do indicador da escala da sequência esperada, mas você não pode atribuir um re-pai do cabeçalho de pacote de informação este pacote, a seguir o pacote está deixado cair e marcado como perdido recebido.
20. **A criptografia é apoiada com dMLP?** Não.
21. **Nós apoiamos a compressão de cabeçalhos PFC?** Não, não no trajeto distribuído. O roteador da ponta oposta não é recomendado configurar a compressão de cabeçalhos PFC porque nós caímos de volta ao modo não-distribuído se nós recebemos quadros ou pacotes comprimidos do encabeçamento. Se você quer continuar a executar o dMLP, a compressão de cabeçalhos PFC deve ser desabilitada no ambas as extremidades.
22. **O compactação do software é apoiado com dMLP?** Não, porque o compactação do software não trabalhará no trajeto distribuído.
23. **A fragmentação é apoiada no lado de transmissão?** Não com baunilha dMLP. Não há nenhuma edição com recepção de fragmentos com baunilha dMLP, mas no lado de transmissão, a fragmentação não acontece. A fragmentação do lado de transmissão é apoiada quando a **intercalação do multilink de PPP** é configurada na relação dMLP.
24. **Podemos nós sibilar os enlaces membros de um conjunto MLP?** Não, você não pode configurar um endereço IP de Um ou Mais Servidores Cisco ICM NT nos enlaces membros.
25. **Há alguma dependência no link MTU e no tamanho do fragmento MLP?** Não. O tamanho do MTU não tem nada fazer com o tamanho do fragmento MLP, a não ser a limitação óbvia que um fragmento MLP, como nenhum outro quadro, não pode exceder os tamanhos do MTU dos enlaces serial.

26. **É possível configurar dois conjuntos MLP entre um único par de Roteadores?** Sim, é possível. Contudo, isto podia conduzir ao Balanceamento de carga danificado. Pode ser útil em testbeds, para simular mais de dois Roteadores que usam apenas dois Roteadores, mas não tem nenhum valor óbvio do mundo real. Todos os links que vão a um par comum *devem* ser postos no mesmo pacote. Por definição, um pacote é o grupo de links que estão indo a um peer particular. Um “par” é identificado pelos valores username e de discriminador de ponto final que oferece durante o LCP e as fases de autenticação. Se você está tentando criar pacotes múltiplos entre dois Roteadores, a seguir significa que você está tentando fazer cada roteador masquerade como sendo mais do que um único par a suas contrapartes. Devem identificar-se apropriadamente.
27. **Podem os enlaces membros ter algoritmos de enfileiramento diferentes?** Todos os mecanismos de filas relativos a uma necessidade do pacote de ser aplicado a nível do pacote e não a nível do enlace membro. Contudo, configurar um algoritmo da fila não deve afetar como os pacotes são comutados fora do pacote.
28. **Por que o TX-quique-limite está ajustado a 26 como o padrão para enlaces membros para um conjunto multilink quando o dMLP é permitido em um Cisco 7500?** Para toda a interface serial da largura de banda T1/E1, o TX-fila-limite é ao redor 4 ou 5. Quando você está empacotando T1s/E1s junto no multilink, a largura de banda aumentaria para o pacote. Porque comutar ocorreria baseado na largura de banda da relação MLP, você precisa de aumentar o TX-fila-limite de enlaces membros. Somente um dos enlaces membros, chamado o link principal, é usado comutando, conseqüentemente, sua necessidade do TX-fila-limite de ser aumentado. Também, este valor é empírico escolhido após o teste e então o ajustamento a este valor. Geralmente, as disposições não têm mais de 4 aos links 6 T1/E1 em um pacote. Um valor de 26 pode cobrir 6 a 8 links T1/E1 perfeitamente, e daqui este valor foi escolhido.
29. **Que é retardo de diferencial e seu valor na implementação de dMLP?** o dMLP apoia um retardo de diferencial da Senhora 30. Isso significaria se um fragmento está recebido em um momento T e é fora de serviço (esperando um número de sequência 100, mas nos receberam 101). Se o número de sequência 100 não é recebido até a Senhora T+30, 100 estariam declarados perdidos e se você pode começar processar de 101, você faria aquele. Caso que você não pode começar com 101 (se é um fragmento médio), você procuraria o fragmento que tem o fragmento do começo (por exemplo, 104) e o começo de lá.
30. **Que acontece quando os pacotes são fragmentados a nível IP com multilink em 7500?** Se os pacotes são fragmentados a nível IP, a seguir estão transportados sem remontagem nos saltos intermediários mas remontados no roteador de destino.
31. **Que acontece quando os pacotes são fragmentados a nível MLP em 7500?** Se os pacotes estão fragmentados a nível MLP e se os pacotes remontados são maiores do que o MRRU, a seguir os pacotes estão deixados cair no multilink. A fragmentação do lado de transmissão é apoiada no dMLP somente com dLFI. Os pacotes estão fragmentados a nível MLP somente se o packet_size é maior do que o frag_size e menos do que o MRRU. Se os pacotes mais do que o MRRU são enviados e se não está fragmentado a nível IP, a seguir a outra extremidade deixa cair todo o tamanho que de pacotes não são fragmentados a nível MLP porque os pacotes são mais do que o MRRU.
32. **Como o MRRU é calculado?** O MRRU é calculado de acordo com estas preferências: Para os enlaces membros novos que entram, o MRRU é negociado outra vez a nível LCP de acordo com o MRRU configurado nos enlaces membros. O valor configurado na interface de link com o comando `ppp multilink mrru interface`. Se não configurado, o valor herdado da

configuração do **comando ppp multilink mrru** na relação do pai. Se ambos os valores estão presentes, o valor da interface de link tem a precedência. O padrão MRRU é de 1524.

Debug realces

Estes realces serão pegados no futuro. O planejamento não é terminar ainda.

- Permita o **comando debug frame-relay multilink** no LC.
- Aumente a corrente de debug CLI pela relação e o número especificado de pacotes.
- Para o dDDR, a funcionalidade de QoS não é apoiada ainda. Isto pode ser pegado somente com caso de negócios apropriado.

Informações Relacionadas

- [Dialer CEF](#)
- [Configuração de DDR Peer-para-Peer com Perfis de Discador](#)
- [MPLS — Apoio do Multilink PPP](#)
- [Multilink Point-to-Point Protocol distribuído para Cisco 7500 Series Routers](#)
- [Frame Relay distribuído do Multilink \(FRF.16\)](#)
- [Fragmentação de link distribuído e intercalação sobre linhas alugadas](#)
- [Links de VoIP por PPP com qualidade de serviço \(LLQ / prioridade IP RTP, LFI, cRTP\)](#)
- [Pesquisando defeitos TechNotes - Cisco 7500 Series Router](#)
- [Página de suporte de produto de roteador - Cisco Systems](#)
- [Suporte Técnico e Documentação - Cisco Systems](#)