

75xx/76xx路由器 — 配置和驗證分散式功能

目錄

[簡介](#)

[必要條件](#)

[需求](#)

[採用元件](#)

[慣例](#)

[分散式功能](#)

[分散式MLPPP](#)

[分散式LFI](#)

[dMLP和dLFioLL之間的差異](#)

[分散式MLFR](#)

[分散式DDR](#)

[分散式功能前提條件和限制](#)

[捆綁包和鏈路的數量以及記憶體要求](#)

[7600 SIP線卡上的硬體和軟體MLPPP或MLFR](#)

[封包的生命週期](#)

[Rx資料路徑](#)

[Tx資料路徑](#)

[重組](#)

[配置、驗證和調試分散式功能](#)

[配置和驗證dMFR](#)

[配置和驗證dMLP/dLFioLL](#)

[配置和驗證dLFioFR和dLFioATM](#)

[配置和驗證dDDR](#)

[調試dMLP和dDDR](#)

[常見問題](#)

[偵錯增強功能](#)

[相關資訊](#)

簡介

本檔案將幫助您瞭解、設定和驗證以下功能：

- 分散式多重連結點對點通訊協定(dMLP)
- 分散式連結分段和交錯(LFI)
- 使用租用線路的分散式LFI(dLFioLL)
- 透過訊框中繼的分散式LFI(dLFioFR)
- 使用ATM的分散式LFI(dLFioATM)
- 分散式MLP(dMLP)和dLFioLL之間的差異

- 分散式多重連結訊框中繼(dMLFR)
- 分散式按需撥號路由(DDR)

必要條件

需求

本文檔的讀者應瞭解Cisco 7500/7600/6500的分散式功能。

採用元件

本文中的資訊係根據以下軟體和硬體版本：

- 所有Cisco 7500和7600平台**注意**：本文檔中的資訊也適用於6500平台。
- 相關的Cisco IOS®軟體版本，下表列出了這些版本：

支援每個分支機構和平台的分散式功能

功能	支援的埠介面卡 (PA) ¹	7500平台		7600平台	
		主要 Cisco IOS 軟體版本	Cisco IOS版本 (臨時)	主要 Cisco IOS 軟體版本	Cisco IOS軟體版本 (中期)
dMLP	Chan-PA PA-4T+ PA-8T	12.0T 12.0S 12.1T 12.2 12.2T 12.3 12.3T 12.2S 12.1E ²	12.0(3)T及更高版本 12.0(9)S及更高版本	12.2SX 12.1E ²	—
dLFloLL	Chan-PA PA-4T+ PA-8T	12.2T 12.3 12.3T 12.0S	12.2(8)T及更高版本 12.0(24)S及更高版本	12.2SX	12.2(17)S XB及更高版本
dLFlo	Chan-	12.2	12.2(4)T3及更高	12.2	12.2(17)S

FR	PA-4T+PA-8T	T 12.3 12.3 T	版本	SX	XB及更高版本
dLFlo ATM	PA-A3 PA-A6	12.2 T 12.3 12.3 T	12.2(4)T3及更高版本	12.2 SX	12.2(17)S XB及更高版本
dMLFR	Chan-PA-4T+PA-8T	12.0 S 12.3 T	12.0(24)S及更高版本 12.3(4)T及更高版本	12.2 SX	12.2(17)S XB及更高版本
dMLP上的QoS	Chan-PA-4T+PA-8T	12.0 S 12.2 T 12.3 12.3 T	12.0(24)S及更高版本 12.2(8)T及更高版本	12.2 SX	12.2(17)S XB及更高版本
dMLP上的MPLS on dLFlo LL	Chan-PA-4T+PA-8T	12.2 T 12.3	12.2(15)T11及更高版本 12.3(5a)及更高版本	12.2 SX	12.2(17)S XB及更高版本
分散式DDR	PA-MC-xT1 PA-MC-xE1 PA-MC-xTE1 PA-MCX-xTE1	12.3 公噸	12.3(7)T及更高版本	—	—

注意：請注意以下資訊：

- 這些PA支援分散式功能：CT3IPPA-MC-T3PA-MC-2T3+PA-MC-E3PA-MC-2E1PA-MC-2T1PA-MC-4T1PA-MC-8T1PA-MC-8E1PA-MC-8TE1+PA-MC-STM-1
- Cisco IOS軟體版本12.1E在7500和7600平台上都支援這些功能。

本文中的資訊是根據特定實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除（預設）的組態來啟動。如果您的網路正在作用，請確保您已瞭解任何指令可能造成的影響。

慣例

如需文件慣例的詳細資訊，請參閱[思科技術提示慣例](#)。

分散式功能

本檔案將說明以下功能：

- 分散式MLP
- 分散式LFI
- 通過租用線路的分散式LFI
- 透過訊框中繼的分散式LFI
- 使用ATM的分散式LFI
- dMLP和dLFIoLL之間的差異
- 分散式MLFR
- 分散式撥號器
- 支援分散式功能的平台和線卡

分散式MLPPP

分散式多連結點對點通訊協定(dMLP)功能允許您將思科7500或7600系列路由器上的線路卡(VIP、FlexWAN)中的全數或部分T1/E1線路合併到具有多個連結的合併頻寬的套件組合中。使用分散式MLP捆綁包來完成此操作。使用者可以選擇路由器中的捆綁包數量和每個捆綁包的鏈路數量。這允許使用者增加網路鏈路的頻寬，使其超過單個T1/E1線路的頻寬，而無需購買T3線路。在非dMLP中，所有資料包都由路由處理器(RP)交換；因此，此實施會影響RP的效能，因為只有幾條運行MLP的T1/E1線路的CPU使用率高。使用dMLP時，路由器上可以處理的捆綁包總數會增加，因為資料路徑是由線卡CPU和記憶體處理和限制的。dMLP允許您捆綁部分T1/E1，從DS0(64 Kbps)開始。

分散式LFI

dLFI功能支援在低速幀中繼和ATM虛擬電路(VC)上以及在租用線路上傳輸即時流量（如語音）和非即時流量（如資料），而不會對即時流量造成過度延遲。

此功能是通過訊框中繼、ATM和租用線路使用多重連結PPP(MLP)實現的。該功能將大型資料包分段為一系列較小的片段，以使對延遲敏感的即時資料包和非即時資料包共用同一鏈路。然後將片段與即時封包交錯。在鏈路的接收端，將重組片段，並重建資料包。

dLFI功能通常適用於通過分散式低延遲隊列（例如語音）傳送即時流量但存在頻寬問題的網路。由於傳輸大型、對時間不太敏感的資料包，這會延遲即時流量。您可以在這些網路中使用dLFI功能，將大型資料封包分解成多個區段。然後，可以在資料分組的這些段之間傳送即時業務分組。在此案例中，即時流量在等待低優先順序資料封包穿越網路時，不會遇到長時間延遲。在鏈路的接收端重組資料包，以便完整地傳送資料。

根據使用`ppp multilink fragment-delay n`命令配置的多鏈路捆綁上的片段延遲來計算鏈路的片段大小，其中：

$$\text{fragment size} = \text{bandwidth} \times \text{fragment-delay} / 8$$

此片段大小僅表示IP負載。不包括封裝位元組（片段大小=權重 — 封裝位元組）。術語「weight」和「fragment size」在RP上`show ppp multilink`命令的輸出中顯示。如果未配置片段延遲，則計算最

大片段延遲為30的預設片段大小。

注意：對於具有不同頻寬的鏈路，所選擇的片段大小基於具有最低頻寬的鏈路。

通過租用線路的分散式LFI

dLFIoLL功能將分散式鏈路分段和交織功能擴展到租用線路。在多鏈路組介面上使用**ppp multilink interleave**命令配置分散式LFI。建議您在頻寬小於768 Kbps的多鏈路介面上使用分散式LFI。這是因為頻寬大於768 Kbps的1500位元組封包的序列化延遲在可接受的延遲限制內，不需要進行分段。

透過訊框中繼的分散式LFI

dLFIoFR功能是透過訊框中繼的多連結PPP(MLPoFR)功能的延伸。MLP用於分段。此功能與FRF.12類似，後者支援分段，並可通過低延遲隊列來交織高優先順序資料包。

虛擬模板上需要**ppp multilink interleave**命令才能在相關聯的虛擬訪問介面上啟用交錯。除了在串列介面上啟用分散式CEF交換之外，此命令也是分散式LFI正常工作的先決條件。

注意：除非您使用幀中繼到ATM網際網路，否則建議您使用FRF.12而不是dLFIoFR，因為使用FRF.12的頻寬利用率更高

使用ATM的分散式LFI

dLFIoATM功能是使用ATM的多連結PPP(MLPoATM)功能的擴展。MLP用於分段。

虛擬模板上需要**ppp multilink interleave**命令才能在相關聯的虛擬訪問介面上啟用交錯。除了在串列介面上啟用分散式CEF交換之外，此命令也是分散式LFI正常工作的先決條件。

使用dLFIoATM時，選擇使封包適合ATM信元的片段大小非常重要，如此一來，就不會在ATM信元中造成不必要的填充。例如，如果選定的片段大小為124位元組，這意味著124位元組的IP負載最終將變為 $124 + 10$ (MLP標頭) + 8 (SNAP標頭) = 142位元組。必須注意的是，第一個分段將以 $124 + 10 + 2$ (第一個分段PID標頭大小) + $8 = 144$ 位元組結尾。這表示此封包將使用三個ATM信元來傳輸負載，因此，將使用封裝的信元效率最高。

dMLP和dLFIoLL之間的差異

dMLP不支援傳輸端分段，而dLFIoLL支援分段。

注意：用於語音流量的多鏈路捆綁中的多個鏈路的交織和分段並不保證按順序接收通過捆綁中的多個鏈路接收的語音流量。語音的正確順序將在上層處理。

分散式MLFR

分散式MLFR功能將基於幀中繼論壇多鏈路幀中繼UNI/NNI實施協定(FRF.16)的功能引入支援線卡的Cisco 7500和7600系列路由器。分散式MLFR功能提供了一種經濟高效的方式，可為特定應用增加頻寬，因為它允許將多個串列鏈路聚合為單個頻寬捆綁。幀中繼網路中的使用者到網路介面(UNI)和網路到網路介面(NNI)支援MLFR。

捆綁包由多條串列鏈路組成，稱為捆綁包鏈路。套件組合中的每個套件組合連結都對應一個實體介面。幀中繼資料鏈路層看不到捆綁鏈路，因此不能在這些介面上配置幀中繼功能。要應用於這些鏈

路的常規幀中繼功能必須在捆綁介面上配置。對等裝置可以看到捆綁鏈路。

[分散式DDR](#)

分散式DDR功能允許撥號器介面上的分散式交換。如果沒有此功能，所有撥入流量都必須傳送到主機進行交換。有了它，只有控制資料包會被傳送到RP，而交換決策則是在建立連線之後在VIP上完成的。

只有PPP封裝支援傳統撥號器配置和撥號器配置檔案配置。還支援Dialer介面上的MLP。撥號器介面上的分散式交換不支援QoS。

[分散式功能前提條件和限制](#)

[必要條件](#)

以下是所有這些分散式功能的常規先決條件：

- 必須全域性啟用分散式思科快速轉發(dCEF)交換。
- 成員串列介面必須啟用dCEF交換，該介面是MLP捆綁包的一部分。
- 必須在dLFloFR和dLFloATM介面的物理鏈路上啟用dCEF交換。
- 分配LFloFR和LFloATM需要交錯配置。
- 在虛擬模板介面上為dLFloFR和dLFloATM介面配置所需的頻寬。
- 在RP上啟用PPP調試時，您可能觀察到_{MLP}：處理器(RSP)上的錯誤介面消息。由於此消息令人困惑且不受歡迎(尤其是如果該消息用於Cisco Discovery Protocol(CDP)資料包)，因此您必須在捆綁包的成員鏈路上配置**no cdp enable**。
- 套件組合的所有成員連結均應啟用keepalive。

[限制](#)

以下是所有這些分散式功能的一般限制：

- T1和E1線路不能混合在套件組合中。
- 支援的最大差分延遲為30毫秒。
- 套件組合中的所有線路必須位於同一個連線埠配接器(PA)上。
- 不支援硬體壓縮。
- VIP或FlexWAN CEF僅限於IP;所有其他協定將傳送到RSP。
- 傳輸端不支援dMLP和dMLFR分段。
- dLFI不支援許多較舊的排隊機制。這些機制包括：虛擬模板介面上的公平隊列虛擬模板介面上的隨機檢測自定義隊列優先順序佇列
- 可以使用模組化QoS CLI在流量策略中配置公平佇列、隨機偵測(dWRED)和優先順序佇列。
- 使用dLFloFR或dLFloATM時，每個MLP捆綁僅支援一個鏈路。如果使用dLFloFR或dLFloATM時，MLP捆綁包中使用了多個連結，則dLFI將自動禁用。在租用線路上使用dLFI時，可在MLP捆綁包中配置多條鏈路使用dLFI。
- 使用dLFloATM時，僅支援aal5snap和aal5mux。不支援封裝aal5nlpid和aal5ciscopp。
- 僅支援IP語音；dLFI功能不支援幀中繼語音和ATM語音。
- 當使用以下功能組合時，不應在多鏈路介面上配置壓縮即時協定(CRTP)配置：啟用LFI的多鏈路介面多重連結套件組合包含多個成員連結在多鏈路介面上啟用具有優先順序功能的QoS策略通過dMLP和dLFI配置，優先順序資料包不攜帶MLP報頭和序列號，並且MLP將在所有成員鏈路之

間分發優先順序資料包。因此，由CRTP壓縮的資料包到達接收路由器的順序可能會混亂。這將禁止CRTP對資料包報頭進行解壓縮，並強制CRTP丟棄資料包。

建議

建議套件組合中的成員連結具有相同的頻寬。如果向捆綁包新增不相等頻寬鏈路，將導致大量資料包重新排序，從而導致捆綁包總吞吐量下降。

建議將VIP2-50 (具有8 MB SRAM) 或更高版本用於這些分散式功能。

捆綁包和鏈路的數量以及記憶體要求

請參閱[適用於Cisco 7500系列路由器的分散式多重連結點對點通訊協定](#)。

7600 SIP線卡上的硬體和軟體MLPPP或MLFR

MPL和MLFR可以基於軟體或硬體。在基於硬體的MPL或MLFR中，Freedm提供多鏈路功能，MPL報頭由Freedm Chip新增。在基於軟體的MPL或MLFR中，SIP線卡CPU提供多鏈路功能 (類似於VIP和FlexWAN實施) 。

以下是運行基於硬體的MPL或MLFR的限制和條件。

- 每個線卡最多只能有336個捆綁包，每個安全狀態評估(SPA)(Freedm)最多只能有168個捆綁包。
- 每個捆綁包最多只能有12個DS1/E1。
- 所有連結都應屬於同一個SPA(Freedm)。
- 套件組合中的所有連結均應以相同速度運作。
- TX片段大小可以是128、256或512。CLI配置的片段大小對映到最接近的支援片段大小。

```
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
```
- RX片段大小可以是1到9.6 KB。
- 不支援Cisco專有格式(MLFR)。

在硬體LFI中，如果捆綁包中只有一個鏈路，而且該鏈路是DS1/E1，則Freedm將執行分段和交錯。

`show ppp multilink`的輸出顯示了硬體實施是否正在運行。

```
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.
```

如果多鏈路是基於軟體的，則`show ppp multilink`輸出中不會有Multilink in Hardware。

封包的生命週期

Rx資料路徑

1. 驅動程式收到的資料包。
2. 封裝已檢查：如下所示基本封裝：在dMLP中，輸入介面的封裝型別是ET_PPP。在dMLFR中，輸入介面的封裝型別是ET_FRAME_RELAY。在dLFloLL中，輸入介面的封裝型別是ET_PPP。在dLFloFR中，輸入介面的封裝型別是ET_FRAME_RELAY。在dLFloATM中，輸入介面的封裝型別是ET_ATM。在dDialer中，封裝型別是ET_PPP。其他封裝處理：對於ET_PPP，將嗅探NLPID。對於dMLP，NLPID為MULTILINK。對於dLFloLL，需要考慮兩個因素：VoIP資料包 — 這些資料包沒有MLP報頭，並且具有表示IP的NLPID。Data Packets - NLPID為MULTILINK。對於dDialer，資料包將不具有MLP報頭並且具有NLPID以指示IP。**注意**：在這種情況下，您可以配置dCRTP（分散式壓縮即時協定）。如果是，則在進一步處理之前解壓縮信頭。
3. 對於ET_FRAME_RELAY，如果接收資料包的鏈路配置為dMLFR，則資料包將處理為dMLFR
4. dLFloFR和dLFloATM的封裝型別分別為ET_FRAME_RELAY和ET_ATM；但其中有一個PPP報頭。PPP報頭與dLFloLL一樣，將指示資料包是語音資料包還是資料包。如果配置了dCRTP，則會在進一步處理之前解壓縮報頭。立即交換語音資料包。分段的資料包在交換之前必須重組。使用ET_PPP時，可能會遇到PPP鏈路資料包；而使用ET_FRAME_RELAY，可能會遇到MLFR控制資料包。所有這些控制資料包都將傳送到RP進行處理。
5. 根據上述解碼，檢查資料包所需的交換型別。鏈路型別將確定資料包是IP交換還是MPLS交換。然後，將分組賦予相應的交換功能。
6. 結合分散式功能捆綁後，IP turbo快速交換向量被竊取。這是因為成員連結上已收到封包；但是，必須將其視為在捆綁上接收。您還需要檢查是否有被傳送到主機的控制資料包。主要在dMLFR中，存在本地管理介面(LMI)資料包，這些資料包不是MLFR控制資料包。對於這些情況，使用dLCI編號空間的不同部分。每當dLCI解碼為落入此空間時，資料包都會被傳送到主機，因為它被識別為LMI資料包。VoIP資料包（在低延遲隊列中排隊）僅在未新增MLP報頭的情況下被關閉。當接收到分段的資料分組時，分散式功能可以接收和重組分組。重組過程將在後續章節中介紹。如果資料包必須採用標籤交換，則會在dMLP中傳遞給標籤交換常式。否則，如果要進行IP交換，則會將其傳遞到IP交換常式。**注意**：所有非IP資料包都會在dMLFR中傳送到主機。
7. IP:IP交換功能對於所有資料包都是通用的。它主要做三件事：如果已設定任何功能，請對封包進行必要處理。此外，使用分散式撥號器時，在收到「相關資料包」時在此處進行空閒計時器更新。請參閱[dialer idle-timeout\(interface\)](#)、[dialer fast-idle\(interface\)](#)和[Configuring a Dialer Profile](#)，以取得閒置計時器設定引數的詳細資訊。在75xx路由器上，鄰接將指示輸出介面的tx_acc_ptr。如果輸出介面是虛擬訪問介面，則tx_acc_ptr為NULL。在這種情況下，請修復封裝並從fib hwidb取得tx_acc_ptr。在dLFloFR和dLFloATM中，此查詢和封裝修復是必需的。在dLFloLL中，連結視為多重連結套件組合的一部分。**注意**：此處調整封包的TTL，並檢查IP分段。所有資料包的mci_status設定為RXTYPE_DODIP。
8. 做出交換決策後，資料包即可從介面發出。檢查介面以確定它是否支援本地交換。若有，則直接透過fastsend傳送。否則，將嘗試對它進行路由快取交換。請注意，如果為介面配置了QoS，則本地交換向量會被QoS竊取。HQF會將資料包入隊並進一步處理該資料包，最後將其從介面傳送出去。dLFI的情況就是這樣。對於dLFI，設定分段和交錯。QoS處理分段常式的呼叫，並將分段的資料包與將在優先順序隊列中排隊的語音資料包進行交織（如果已配置LLQ）。這可確保VoIP資料包不會因通過鏈路傳輸大量資料包而出現延遲。

Tx資料路徑

`vip_dtq_consumer` 獲取資料包並獲取介面編號，然後從中獲取 `idb`。與 `idb` 對應的快速傳送常式稱為：

i) 快速傳送

1. 在 dMFR 中，從與 `if_index` 和 `fr_info` 匹配的表中檢索 `fr_info` 結構。控制資料包剛剛發出。幀頭將提供 dLCI，這將幫助您確定這是 LMI 資料包還是資料包。幀頭中的 `dLci` 欄位被 `dmfr` 序列號覆蓋。LMI 和資料包使用單獨的序列號。**注意：**單獨的 dLCI 使用單獨的序列號。
2. 在 dMLP 中，控制資料包的傳送優先順序設定為高。對於資料包，如果配置了 dCRTP，則會壓縮報頭。新增包含排序資訊的 VIP MLP 標頭，並將其從成員連結傳送出去。
3. 在 dLFI 中，HQF 會攔截要通過介面傳送的資料包。如果是語音資料包，語音資料包將被置於優先順序隊列（如果已配置 LLQ）中，並且不使用 MLP 封裝從介面發出。對於資料包，它呼叫 dLFI 分段代碼，該代碼將片段返回到 QoS 代碼，然後將這些片段與優先順序流量交錯，以滿足語音流量的延遲要求。此外，如果已配置 dCRTP，則僅壓縮語音資料包的報頭。資料包報頭保留原樣。
4. 在 dDialer 中，對資料包進行分類，以便在傳送資料包之前重置輸出鏈路的空閒計時器。如果多個鏈路繫結到同一個撥號器，則會在選擇輸出鏈路後完成此操作。未向撥號器封包新增標頭。因此，撥號器介面不支援資料包的排序和重組。

註：在具有多個鏈路的 dMLP、dDialer、dMLFR 和 dLFI 中，轉發流量的物理鏈路取決於鏈路的擁塞。如果鏈路擁塞，請移至下一個鏈路，以此類推。（dMLFR、不帶 QoS 的 dMLP 和 dDialer 功能也會根據鏈路上放置的位元組數選擇鏈路。如果當前鏈路已經以循環方式傳輸其位元組配額，它將選擇下一個鏈路。此配額由鏈路的 `frag_bytes` 決定。對於 Dialer 成員介面，`frag_bytes` 設定為介面頻寬的預設值。）

注意：在出口 VIP 介面上的 HQF 配置中，HQF 會竊取 `dtq_consumer` 向量。到達出口 VIP 的資料包 DMA 首先通過 HQF 檢查。如果在輸出介面上配置了 QoS，則 HQF 會在資料包從介面快速傳送出去之前啟動以處理資料包。

重組

普通 dDialer 介面不支援重組和排序。要在撥號器介面上啟用此功能，必須配置通過撥號器介面的 MLP。如果這樣做，則 Rx 和 Tx 路徑與 dMLP 路徑相同。收到資料包時，將根據預期序列號檢查序列號。

- 如果序列號匹配：如果該封包是未分段封包，則不需要重組。繼續進一步切換步驟。如果封包是片段，則檢查開始位和結束位，並在收到片段時構建封包。
- 如果序列號不匹配：如果序列號位於序列號預期視窗內，則將其放入已排序的「未分配的片段清單」中。稍後，如果沒有收到預期的序列號，則會檢查此清單，以防資料包儲存在此處。如果序列號不在視窗中，請丟棄該序列號並報告「收到丟失的片段」。如果在等待此資料包時出現超時，則重新對接收方進行同步，並從接收的下一個資料包重新開始。

在所有這些情況下，正確排序的資料包流都會從該介面發出。如果收到片段，則會形成完整資料包，然後傳送出去。

配置、驗證和調試分散式功能

本節介紹 `show` 和 `debug` 命令，這些命令可用於驗證和調試每個分散式功能。

配置和驗證 dMFR

MFR示例配置

```
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
```

注意：MFR介面類似於另一個FR介面，因此支援大多數FR配置。

```
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
```

驗證RP上的MFR捆綁狀態

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
```

這表示兩個介面新增正確，其中一個介面尚未協商MLFR LIP消息。

要獲取有關MFR捆綁包和成員連結的更多資訊，請發出以下命令：

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
```

[MFR Debug命令](#)

這些調試對於排除連結未新增到捆綁包中的問題非常有用。

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

注意：如果未指定特定的MFR介面或串列介面，則會啟用所有MFR鏈路的調試。如果路由器具有大量的MFR鏈路，這會造成極大的負擔。

若要調試RP接收的MFR資料包以及調試MFR控制活動，此調試非常有用：

```
debug frame-relay multilink
```

注意：在繁忙的流量下，這可能會使CPU不堪重負。

[驗證LC上的dMLFR套件組合狀態](#)

```
show frame-relay multilink
```

注意：目前，LC上無法提供此功能，但很快就會新增它。在此之前，請使用 `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

```

[配置和驗證dMLP/dLFIoLL](#)

[多重連結PPP組態](#)

```

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

```

串列介面下的配置示例：

```

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
!

```

註： `ppp chap hostname M1` 命令並不真正表示已啟用CHAP身份驗證。此命令中的字串M1用作終端鑑別器，僅當同一兩台路由器之間將存在多個多鏈路捆綁時才需要。在這種情況下，屬於某個捆綁包的所有鏈路都應具有相同的端點鑑別器，而屬於不同捆綁包的兩個鏈路不應具有相同的端點鑑

別器。

[可選配置引數](#)

[no] ppp multilink interleave

這樣可對多鏈路捆綁包啟用交錯。此功能與模組化QoS CLI配合使用。高優先順序資料包將在不增加MLP序列和報頭的情況下傳輸，而其他資料包將進行分段並使用MLP序列和報頭傳輸。

注意：當啟用有多個鏈路的交織時，高優先順序流量可能會重新排序。啟用或停用交錯，需要重設套件組合，才能線上路卡中啟動交錯。

```
ppp multilink mrru local value
```

這指定多鏈路上的最大接收單元；多鏈路介面將接受最大為此大小的資料包。此處的大小不包括MLP報頭。

```
ppp multilink mrru remote value
```

這指定遠端應支援的最小MRRU。如果遠端MRRU小於此值，則套件組合交涉將失敗。

```
ppp multilink fragment delay seconds
```

這指定允許的資料片段導致的延遲（毫秒）。換句話說，延遲值用於計算允許的最大片段大小。分散式實作與Cisco IOS實作在以下方面不同：

1. 除非啟用交錯，否則不會執行分段。
2. 對於頻寬變化的鏈路，所選擇的片段大小基於最小頻寬介面。

```
ppp multilink fragment disable
```

此命令不會在分散式實施中新增任何功能。僅當啟用交織時才進行分段；並且，當啟用交織時，`ppp multilink fragment disable`命令會被忽略。

[驗證RP上的dMLP捆綁狀態](#)

```
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. 當套件組合處於分散模式時，會顯示在show ppp multilink的輸出中：如果不是，則由於某種原因不會分發捆綁包。

2. 線上卡上配置並啟用ppp multilink interleave時，show ppp multilink輸出包括dLFI，其中：
 : Fragmented — 指示已傳輸和接收的片段的計數。Unfragmented — 表示未分段就被傳輸或接收的資料包數量。Reassembled — 表示已重組的完整資料包數。未啟用交織時，輸出如下所示：

```

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

```

上例中的片段大小為760位元組。

[驗證LC上的dMPLP套件組合狀態](#)

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

```

使用dMFR時，序列號基於每個dLCI維護，捆綁中的序列號用於LMI dLCI。

欄位	說明
dmlp_ipc_config_count	LC接收的用於多鏈路或MLFR配置的IPC消息數
dmlp_bundle_count	LC中MLP和MLFR捆綁包的數量
dmlp_ipc_fault_count	在LC中導致故障的配置消息數。應為0;如果它非零，則可能存在問題。
	指示標籤交換中使用的idb to tag_optimum_fs和idb to ip2tag_optimum_fs向量。
board_encap	指示7500平台中有通道化鏈路時用於新增2位元組的板封裝的board_encap向量。如果鏈路包含非通道化介面，則應為NULL。
max_particles	可在重組緩衝區中保留的最大粒子數
mrru	在不考慮MLP封裝的情況下接受的最大資料包大小。不適用於MLFR介面。
seq_window_size	序列號的最大視窗大小
working_pak	指示正在重組的當前pak。 NULL，如果沒有。
working_pak_cache	指向用於重組的靜態pak的指標。當套件組合收到第一個非完整封包時，系統會分配此封包。
una_frag_list	重組隊列中的第一個條目。如果條目不是NULL且未更改，則表明計時器未運行軟體問題。
una_frag_end	重組隊列中的最後一個條目
rcved_end_bit	表示套件組合已收到結束位元，因此正在搜尋開始位元。
is_lost_frag	

	如果片段宣告丟失，則為true。收到具有預期序列的片段時，將清除此項。
resync_count	指示接收器與發射器不同步，且必須從最後接收的序列片段開始重新同步的次數。
	指示發生了重組超時，並且正在處理來自重組隊列的資料包。
timer_start	已啟動重組計時器的次數
timer_running	指示重組計時器是否正在運行。
timer_count	指示重組計時器已過期的次數。
next_xmit_link	傳輸下一個資料包的鏈路
	表示成員存在的位欄位。
	並非在所有分支中使用的欄位。指示哪些成員連結未擁塞。
dmlp_orig_pak_to_host	用於將資料包傳送到RP的向量。
dmlp_orig_fastsend	在MLP或MLFR修改驅動程式的fastsend之前，原始驅動程式fastsend。
	丟失的片段數量（接收方沒有收到這些片段）。當更新傳送到主機時，會定期清除此項。
	超出預期順序接收的片段數。當更新傳送到主機時，會定期清除此項。
	由於無法生成完整資料包而丟棄的片段數
	收到被認為是丟失的片段數。這表示連結間延遲大於dMLP重組逾時30毫秒。

配置和驗證dLFioFR和dLFioATM

```

class-map voip
  match ip precedence 3

policy-map llq
  class voip
    priority

int virtual-templ1
  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-templ1

```


!--- Or

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

[驗證RP上的dLFIoFR/ATM捆綁狀態](#)

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
```

注意：只有在虛擬模板下配置ppp multilink interleave時，捆綁包才會變為分發；如果沒有此命令，將不會分發捆綁包。

[驗證LC上的dLFIoFR/ATM套件組合狀態](#)

若要確認dLFI確實在LC上正常工作，請發出以下命令：

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
```

應該有優先順序層和WFQ層。分段將在WFQ枝葉層塊上完成。

配置和驗證dDDR

在全域性配置上啟用ip cef distributed和在撥號器介面上啟用ip route-cache distributed時，會啟用分散式DDR。

```
!--- 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
```

分散式DDR沒有其他特殊配置。進一步配置遵循常規DDR配置。

驗證分散式按需撥號路由

```
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
```

dialer type 告訴我們使用的撥號器型別。ISDN 著傳統撥號器配置，而PROFILE 著撥號器配置檔案配置。dialer state 表示撥號器的當前狀態。未連線的撥號器介面的狀態將為。每當看到關注流量時，Idle timer 都會重置。如果此計時器過期，介面將立即斷開連線。是一個可配置的引數。有關詳細資訊，請參閱[使用撥號程式配置檔案配置對等DDR](#)。

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
```

顯示的變數與dMLP的變數相同。

調試dMLP和dDDR

RP上可用的調試

dDDR

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

發出此命令可對控制路徑功能（如呼叫建立等）進行調試。有關詳細資訊，請參閱[debug dialer events](#)。

```
debug ip cef dialer
```

發出此命令以調試CEF相關的撥號器事件。有關詳細資訊，請參閱[撥號器CEF](#)。

LC上可用的調試

dMLP

控制路徑調試：debug multilink event

資料路徑調試：debug multilink fragments

資料路徑和控制路徑錯誤調試：debug multilink error

在SIP線路卡上調試dMLP

根據CI轉儲資料包：可以根據控制ci和序列ci將資料資料包和控制資料包轉儲到線卡上。

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

可以通過以下方式獲取CI:

```
!--- 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
  encaps 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
對於CT3，必須獲取vc num，該數字可以從show interface serial CT3_interface_name 的輸出中獲取
。
```

現在，可以從SPA控制檯獲取CI資訊。首先使用spa_redirect rp ct3_freedm336命令將SPA控制檯命令的輸出重定向到RP。

spa_ct3_test freedm show linkrec vc 命令會顯示必要的CI資訊。

dmFR

控制路徑調試：debug dmfr event

資料路徑調試：debug dmfr packets

資料路徑和控制路徑錯誤調試：debug dmfr error

根據CI轉儲資料包：參見[dMLP](#)。

dLFI

控制路徑調試：debug dlfi event

資料路徑調試：[debug dlfi fragments](#)

資料路徑和控制路徑錯誤調試：[debug dlfi error](#)

dDDR

沒有特殊的調試命令；應使用[dMLP調試](#)。

在發生dLFloLL的情況下，可能必須同時使用dMLP和dLFI調試。這些調試不是有條件的，因此將觸發所有繫結。

常見問題

1. **什麼是dMLP?**dMLP是分散式多鏈路PPP的縮寫(如[RFC1990](#) 中所述)。分散式平台 (如Cisco 7500系列和7600系列) 支援此功能。dMLP允許您將T1/E1線路 (在Cisco 7500系列路由器上的VIP中或7600系列路由器上的FlexWAN中) 合併到具有多個T1/E1線路合併頻寬的捆綁包中。這樣，客戶就無需購買T3/E3線路，即可將頻寬提高到T1/E1以上。
2. **dMLP中的「分佈」是什麼?**術語「分散式」表示資料包交換由VIP而不是RSP完成。為什麼? RSP交換功能相當有限，而且它還有許多更重要的工作要做。能夠交換資料包的VIP從RSP中解除安裝此活動。基於RSP的Cisco IOS仍然管理鏈路。捆綁包的建立和拆卸由RSP完成。此外，PPP控制平面處理仍由RSP完成，包括所有PPP控制資料包的處理 (LCP、身份驗證和NCP)。但是，一旦建立了捆綁包，MLP資料包的處理將移交給VIP，以便由板載CPU進行交換。dMLP引擎 (在VIP上) 處理所有MLP過程，包括分段、交錯、封裝、多個鏈路間的負載均衡以及入站片段的排序和重組。VIP在7500系統中完成的功能由基於7600的系統中的Flexwan/Enhanced-FlexWAN完成。
3. **如何知道捆綁包是否已分發?**在路由器控制檯上發出show ppp multilink命令：

```
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)
  Sell1/1/0/27:0, since 00:22:46, no frags rcvd
  Sell1/1/0/25:0, since 00:22:46, no frags rcvd
```

!--- Output suppressed.

4. **如果我升級到RSP16或SUP720，我的dMLP效能是否會更好?**否。dMLP (或任何分散式功能) 的交換效能取決於所討論的VIP或FlexWAN。例如，VIP6-80的效能將優於VIP2-50的效能。
5. **哪些個人助理可以使用此功能?** 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 (僅限7500)
6. **單個捆綁包中可配置多少個鏈路?** 這個答案有許多方面。主要瓶頸是線卡 (VIP/FlexWAN/增強型FlexWAN2) 的CPU功率。硬限制是每個捆綁包56條鏈路，但許多情況下由於CPU功率或緩衝區有限，您無法配置這些鏈路 (並且存在大量流量交換)。這些數字基於以下准則 (基於VIP/FlexWAN/增強型FlexWAN2上的CPU和記憶體)：VIP2-50 (帶4MB SRAM) 最大T1 = 12VIP2-50 (帶8MB SRAM) 最大T1 = 16VIP4-80最大T1 = 40VIP6-80最大T1 = 40FlexWAN最大T1 = 稍後將更新增強型FlexWAN最大E1 = 21 E1/托架 (每線卡總計42 E1)
7. **如果我配置3個捆綁包 (每個捆綁包3個T1) 或1個捆綁包(9個T1)，效能是否有變化?** 效能沒有

變化，這一點在實驗室測試中得到了證實。但是，對於單個捆綁包中的大量T1（例如單個捆綁包中的24或28個T1），存在緩衝區不足的問題。強烈建議您在單個捆綁包中的成員鏈路（T1/E1）不要超過8個。

8. **如何確定捆綁包的頻寬？**不配置捆綁包的頻寬。是所有成員鏈路的聚合頻寬。如果套件組合中有4個T1，則套件組合的頻寬為6.144Mbps。
9. **哪個更好？CEF負載均衡還是dMLP？**這個問題沒有簡單的答案。你的需求決定哪一個更好。
MLP的優點：CEF負載均衡僅適用於IP流量。MLP平衡通過捆綁包傳送的所有流量。MLP維護資料包的排序。IP本身可以容忍重新排序，因此這對您來說可能無關緊要；事實上，維持測序的額外成本可能是避免MLP的一個原因。IP適用於可能會亂序傳送資料包的網路，任何使用IP的裝置都應能處理重新排序問題。然而，儘管存在這一事實，但現實情況是，重新排序仍可能構成一個真正的問題。MLP提供到對等系統的單個邏輯連線。多鏈路捆綁包支援QoS。MLP提供動態頻寬功能，因為使用者可以根據當前需求新增或刪除成員鏈路。MLP可以捆綁更多鏈路，而CEF負載平衡限於6個並行IP路徑。每個流CEF負載平衡將任何給定流的最大頻寬限制為一個T1。例如，使用語音網關的客戶可以擁有具有相同源和目標的許多呼叫，因此只能使用一個路徑。**MLP的缺點：**MLP為每個資料包或幀增加額外開銷MLP是CPU密集型；dMLP佔用線路卡CPU資源。
10. **如何在兩台路由器之間配置多個捆綁包？**多重連結根據對等體的名稱和端點鑑別器確定連結將加入的捆綁。要在兩個系統之間建立多個不同的捆綁，標準方法是強制某些連結以不同的方式標識自己。推薦的方法是使用 `ppp chap hostname name` 命令。
11. **是否可以具有來自不同PA的成員連結？**否。如果要運行dMLP，則不支援它。但是，如果從不同的PA新增成員連結，則控制權將授予RSP及其非dMLP。MLP仍在運行，但dMLP的好處已不復存在。
12. **是否可以從兩個機架混合成員連結？**否。如果要運行dMLP，則不支援它。但是，如果從不同的PA新增成員連結，則控制權將授予RSP，而不再是dMLP。MLP仍在運行，但dMLP的好處已不復存在。
13. **是否可以在不同的VIP或FlexWAN間擁有成員鏈路？**否。如果要運行dMLP，則不支援它。但是，如果從不同的PA新增成員連結，則控制權將授予RSP及其非dMLP。MLP仍在運行，但dMLP的好處已不復存在。
14. **是否可以從單個PA跨不同埠擁有成員鏈路？**（例如，來自PA-MC-2T3+的每個CT3埠的一個成員鏈路。）會。只要是來自同一個人助理，就沒有任何問題。
15. **是否可以捆綁T3或E3埠？**否。對於7500/VIP、7600/FlexWAN和7600/FlexWAN2，只有DS0、n*DS0、T1和E1速度允許。**注意：**只有以T1/E1或子速率T1/E1速度配置的成員鏈路才支援分散式MLPPP。通道化STM-1/T3/T1介面也支援T1/E1或子速率T1/E1速度的dMLPPP。以清除通道T3/E3或更高的介面速度配置的成員鏈路不支援分散式MLPPP。
16. **什麼是「重新排序」的片段？**如果收到的片段或資料包與預期序列號不匹配，則reordered計數器將遞增。對於不同大小的資料包，這必然會發生。對於固定大小的資料包，也可能發生這種情況，因為PA驅動程式會處理在一個鏈路上收到的資料包，而不以循環方式執行（在傳輸資料包時，在dMLP中執行此過程）。重新排序並不意味著丟包。
17. **什麼是「丟失」片段？**每當錯誤接收片段或封包且您發現所有連結上接收到錯誤片段或封包時，lost fragments計數器會增加。另一種情況是，當無序片段儲存在清單中，且其達到限制（根據VIP上的SRAM以及分配給套件組合的任何內容而決定）時，會遞增遺失片段計數器，並利用清單中的下一個序號進行處理。
18. **dMLP如何檢測丟失的片段？**序列號：如果您正在等待序列號為N的片段到達，且所有連結都收到序列號高於N的片段，就知道片段N必須丟失，因為無法在同一連結上以更高編號的片段合法到達。逾時：如果等待片段的時間太長，您最終會宣佈片斷已丟失，然後繼續。重組緩衝區溢位：如果您正在等待片段N到達，而同時一些鏈路上的其他片段（序列號高於N），則必須將這些片段暫存到重組緩衝區中，直到片段N顯示。緩衝能力有限。如果緩衝區溢位，則再次宣告片段N已丟失，然後使用緩衝區中的任何內容恢復處理。

19. 什麼是「接收丟失」？收到片段或封包遺失可能有兩個原因：如果收到的片段或資料包超出預期序列範圍視窗，則將資料包標籤為已接收丟失，從而丟棄該資料包。如果收到的片段或資料包在預期序列範圍視窗中，但您無法將資料包報頭分配給此資料包的重新父代，則該資料包將被丟棄，並標籤為已接收丟失。
20. dMPLP是否支援加密？編號
21. 我們是否支援PFC報頭壓縮？不，不在分散式路徑中。建議不要使用遠端路由器來設定PFC報頭壓縮，因為如果我們收到壓縮的報頭幀或封包，會回退到非分散模式。如果要繼續運行dMPLP，必須在兩端禁用PFC報頭壓縮。
22. dMPLP是否支援軟體壓縮？否，因為軟體壓縮在分散式路徑中不起作用。
23. 傳輸端是否支援分段？Vanilla dMPLP除外。使用Vanilla dMPLP接收片段沒有問題，但是在傳輸端，分段不會發生。在dMPLP介面上配置ppp multilink interleave時，支援傳輸端分段。
24. 是否能ping通MLP捆綁的成員鏈路？不能，不能在成員鏈路上配置IP地址。
25. 是否依賴於鏈路MTU和MLP片段大小？不能。MTU大小與MLP片段大小無關，而是MLP片段與任何其他幀一樣不能超過串列鏈路的MTU大小的明顯限制。
26. 是否可以在一對路由器之間配置兩個MLP捆綁包？是的，這是可能的。但是，這可能會導致負載平衡受損。在測試台上使用兩個路由器來模擬兩個以上的路由器是很有用的，但它不具有任何明顯的現實價值。前往共同對等點的所有連結必須放在同一個套件組合中。根據定義，套件組合是前往特定對等點的連結集。「對等體」由它在LCP和身份驗證階段提供的使用者名稱和終端鑑別器值來標識。如果您嘗試在兩個路由器之間建立多個捆綁包，則意味著您試圖使每台路由器偽裝成多個對等路由器。他們必須正確表明自己的身份。
27. 成員鏈路是否可以使用不同的排隊演算法？所有與捆綁相關的排隊機制都需要在捆綁級別應用，而不是在成員鏈路級別應用。但是，配置隊列演算法不應影響資料包從捆綁包中交換的方式。
28. 在Cisco 7500上啟用dMPLP時，為什麼對於多鏈路捆綁的成員鏈路，tx-queue-limit設定為26，作為預設值？對於頻寬T1/E1的任何串列介面，tx-queue-limit約為4或5。在多鏈路中將T1/E1捆綁在一起時，捆綁的頻寬會增加。由於交換將基於MLP介面的頻寬進行，因此需要增加成員鏈路的tx-queue-limit。只有一個成員鏈路（稱為主鏈路）用於交換，因此需要增加其tx-queue-limit。此外，此值是在測試之後選擇的一個經驗值，然後調整到此值。一般來說，部署捆綁包中的多於4到6個T1/E1鏈路。值26可以完全滿足6到8個T1/E1鏈路的需求，因此選擇了此值。
29. dMPLP實現中的差分延遲及其值是什麼？dMPLP支援30毫秒的差分延遲。這意味著如果在時間T處收到片段且其順序打亂（預期序列號為100，但我們收到101）。如果序列號為100，直到T+30毫秒才收到，則系統會宣告丟失100，如果您可以從101開始處理，則會執行上述操作。如果不能從101開始（如果它是中間片段），則應查詢包含begin片段的片段（例如104），並從那裡開始。
30. 當封包在IP層級進行分段（在7500上使用多重連結）時，會發生什麼情況？如果封包在IP層級進行分段，則在傳輸時不會在中間躍點進行重組，而是在目的地路由器上進行重組。
31. 在7500上，當封包在MLP層級分段時，會發生什麼情況？如果封包在MLP層級進行分段，且重組的封包大於MRRU，則封包會在多重連結上捨棄。僅使用dLFI的dMPLP支援傳輸端分段。只有packet_size大於frag_size且小於MRRU時，才會在MLP級別對資料包進行分段。如果傳送的封包超過MRRU且在IP層級未分段，則另一端會捨棄在MLP層級未分段的所有封包大小，因為這些封包大於MRRU。
32. 如何計算MRRU？MRRU是根據以下首選項計算的：對於進入的新成員鏈路，將根據成員鏈路上配置的MRRU再次在LCP級別協商MRRU。使用ppp multilink mrru interface命令在鏈路介面上配置的值。如果未配置，該值是從父介面上的ppp multilink mrru命令配置繼承的。如果存在這兩個值，則鏈路介面值具有優先順序。預設MRRU 1524。

偵錯增強功能

這些增強功能將在將來採用。規劃尚未完成。

- 在LC上啟用**debug frame-relay multilink**命令。
- 增強每個介面和指定數量的資料包的當前調試CLI。
- 對於dDDR，尚不支援QoS功能。這只有在具有適當的業務案例的情況下才能進行。

相關資訊

- [撥號器CEF](#)
- [使用撥號程式配置檔案配置對等DDR](#)
- [MPLS — 多重連結PPP支援](#)
- [適用於Cisco 7500系列路由器的分散式多重連結點對點通訊協定](#)
- [分散式多重連結訊框中繼\(FRF.16\)](#)
- [使用租用線路的分散式連結分段和交錯](#)
- [使用服務品質 \(LLQ/IP RTP優先順序、LFI、cRTP \) 的PPP上的VoIP連結](#)
- [疑難排解技術說明 — Cisco 7500系列路由器](#)
- [路由器產品支援頁面 — Cisco Systems](#)
- [技術支援與文件 - Cisco Systems](#)