75xx/76xx路由器 — 配置并检验分布式功能

目录

简介

本文档帮助您了解、配置和验证以下功能：

- 分布式多链路点对点协议(dMLP)
- 分布式链路分段和交织(LFI)
- 租用线路上的分布式LFI(dLFIoLL)
- 帧中继上的分布式LFI(dLFIoFR)
- 基于ATM的分布式LFI(dLFIoATM)
- 分布式MLP(dMLP)和dLFIoLL之间的差异

- 分布式多链路帧中继(dMLFR)
- 分布式按需拨号路由(DDR)

先决条件

要求

本文档的读者应了解Cisco 7500/7600/6500的分布式功能。

使用的组件

本文档中的信息基于以下软件和硬件版本：

- 所有Cisco 7500和7600平台**注意**：本文档中的信息也适用于6500平台。
- 相关Cisco IOS®软件版本，下表列出：

对每个分支机构和平台的分布式功能支持

功能	支持端口适配器(PA)[1]	7500 平台		7600 平台	
		主要Cisco IOS软件版本	Cisco IOS版本（中期）	主要Cisco IOS软件版本	Cisco IOS软件版本（中期）
dMLP	通道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及以上版本12.0(9)S及以上版本	12.2SX 12.1E[2]	—
dLFIoLL	通道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)SXB及更高版本
dLFIoF	通道	12.2	12.2(4)T3及更高	12.2	12.2(17)

R	PA-4T+ PA-8T	T 12.3 12.3 T	版本	SX	SXB及更高版本
dLFIoATM	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)SXB及更高版本
dMLFR	通道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)SXB及更高版本
dMLP上的QoS	通道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)SXB及更高版本
dMLP上的MPLS dLFIoLL上的MPLS	通道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)SXB及更高版本
分布式DDR	PA-MC-xT1 PA-MC-xE1 PA-MC-xTE1 PA-MCX-xTE1	12.3 T	12.3(7)T及以上版本	—	—

注意：请注意以下信息：

1. 这些PA支持分布式功能：CT3IPPA-MC-T3PA-MC-2T3+PA-MC-E3PA-MC-2E1PA-MC-2T1PA-MC-4T1PA-MC-8T1PA-MC-8E1PA-MC-8TE1+PA-MC-STM-1
2. Cisco IOS软件版本12.1E在7500和7600平台上都支持这些功能。

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您使用的是真实网络，请确保您已经了解所有命令的潜在影响。

规则

有关文件规则的更多信息请参见“ Cisco技术提示规则”。

分布式功能

本文档对以下功能进行了说明：

- 分布式MLP
- 分布式LFI
- 租用线路上的分布式LFI
- 帧中继上的分布式LFI
- 基于ATM的分布式LFI
- dMLP和dLFIoLL之间的差异
- 分布式MLFR
- 分布式拨号器
- 支持分布式功能的平台和线卡

分布式MLPPP

分布式多链路点对点协议(dMLP)功能允许您将Cisco 7500或7600系列路由器上的线卡(VIP、FlexWAN)中的完整或部分T1/E1线路组合为具有多条链路组合带宽的捆绑。您使用分布式MLP捆绑包来执行此操作。用户可以选择路由器中的捆绑包数和每个捆绑包的链路数。这样，用户就可以增加单条T1/E1线路以外的网络链路带宽，而无需购买T3线路。在非dMLP中，所有数据包都由路由处理器(RP)交换；因此，此实施会影响RP的性能，因为只有少数T1/E1线路运行MLP时，CPU使用率较高。使用dMLP时，可在路由器上处理的捆绑总数会增加，因为数据路径由线卡CPU和内存处理和限制。dMLP允许您捆绑部分T1/E1，从DS0(64 Kbps)开始。

分布式LFI

dLFI功能支持在低速帧中继和ATM虚电路(VC)上以及租用线路上传输实时流量（如语音）和非实时流量（如数据），而不会对实时流量造成过大延迟。

此功能使用帧中继、ATM和租用线路上的多链路PPP(MLP)来实现。该功能将大数据包分段为一系列更小的分段，以使延迟敏感的实时数据包和非实时数据包能够共享同一链路。然后，将分段与实时分组交错。在链路的接收端，重组分段并重组数据包。

dLFI功能通常适用于通过分布式低延迟队列（如语音）发送实时流量但存在带宽问题的网络。这会延迟实时流量，因为传输大型、时间敏感性较低的数据包。您可以在这些网络中使用dLFI功能，将大型数据包分解为多个网段。然后，可以在这些数据包段之间发送实时流量数据包。在此场景中，实时流量在等待低优先级数据包通过网络时不会经历长时间延迟。数据包在链路的接收端重组，因此数据传输完整。

链路的分片大小根据多链路捆绑上的分片延迟计算，该捆绑使用**ppp multilink fragment-delay n命令进**行配置，其中：

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

此分段大小仅表示IP负载。它不包括封装字节（分片大小=权重 — 封装字节）。 术语“weight”和“fragment size”在RP上的**show ppp multilink**命令输出中显示。如果未配置分段延迟，则为最大分段延迟30计算默认分段大小。

注意：对于带宽不同的链路，所选分段大小基于带宽最少的链路。

租用线路上的分布式LFI

dLFIoLL功能将分布式链路分段和交织功能扩展到租用线路。在多链路组接口上使用**ppp multilink interleave**命令配置分布式LFI。建议在带宽小于768 Kbps的多链路接口上使用分布式LFI。这是因为带宽大于768 Kbps的1500字节数据包的串行化延迟在可接受的延迟限制内，无需分段。

帧中继上的分布式LFI

dLFIoFR功能是帧中继多链路PPP(MLPoFR)功能的扩展。MLP用于分段。此功能类似于FRF.12，它支持分段，并可通过低延迟队列交错高优先级数据包。

要在关联的虚拟访问接口上启用交织，需要在Virtual-template上使用**ppp multilink interleave**命令。除了在串行接口上启用分布式CEF交换外，此命令也是分布式LFI工作的先决条件。

注：除非您使用帧中继到ATM网际网络，否则建议您使用FRF.12而不是dLFIoFR，因为FRF.12的带宽利用率更高

基于ATM的分布式LFI

dLFIoATM功能是ATM上多链路PPP(MLPoATM)功能的扩展。MLP用于分段。

要在关联的虚拟访问接口上启用交织，需要在Virtual-template上使用**ppp multilink interleave**命令。除了在串行接口上启用分布式CEF交换外，此命令也是分布式LFI工作的先决条件。

使用dLFIoATM时，选择分段大小非常重要，该大小使数据包能够适合ATM信元，这样它们就不会在ATM信元中造成不必要的填充。例如，如果所选分段大小为124字节，这意味着124字节的IP负载最终将变为124 + 10（MLP报头）+ 8（SNAP报头）= 142字节。请注意，第一个分片将以124 + 10 + 2（第一个分片PID报头大小）+ 8 = 144字节输出。这意味着此数据包将使用三个ATM信元来传输负载，因此将最有效地使用封装的信元。

dMLP和dLFIoLL之间的差异

dMLP不支持传输端的分段，而dLFIoLL支持。

注意：在用于语音流量的多链路捆绑中与多个链路一起使用的交织和分段并不保证通过捆绑中多个链路接收的语音流量将按顺序接收。语音的正确顺序在上层处理。

分布式MLFR

分布式MLFR功能将基于帧中继论坛多链路帧中继UNI/NNI实施协议(FRF.16)的功能引入线卡型Cisco 7500和7600系列路由器。分布式MLFR功能提供了增加特定应用带宽的经济高效的方法，因为它允许将多个串行链路聚合到单个带宽捆绑中。帧中继网络中的用户到网络接口(UNI)和网络到网络接口(NNI)支持MLFR。

捆绑包由多个串行链路组成，称为捆绑链路。捆绑中的每个捆绑链路对应一个物理接口。捆绑链路对帧中继数据链路层不可见，因此无法在这些接口上配置帧中继功能。要应用于这些链路的常规帧中继功能必须在捆绑接口上配置。捆绑链路对对等设备可见。

分布式DDR

分布式DDR功能允许在拨号器接口上进行分布式交换。如果没有此功能，所有拨入流量都必须传送到主机进行交换。通过它，仅控制数据包发送到RP，而交换决策在建立连接后在VIP本身上完成。

仅PPP封装支持传统拨号器配置和拨号器配置文件配置。还支持拨号器接口上的MLP。拨号器接口上的分布式交换不支持QoS。

分布式功能必备条件和限制

先决条件

以下是所有这些分布式功能的一般必备条件：

- 分布式思科快速转发(dCEF)交换必须全局启用。
- 必须在成员串行接口上启用dCEF交换，该接口是MLP捆绑包的一部分。
- 必须在dLFIoFR和dLFIoATM接口的物理链路上启用dCEF交换。
- 分配LFIoFR和LFIoATM需要交织配置。
- 在dLFIoFR和dLFIoATM接口的虚拟模板接口上配置所需的带宽。
- 在RP上启用PPP调试时，您可能会看到MLP:换处理器(RSP)上的错误接口消息。 由于此消息令人困惑且不需要，因此必须在捆绑的成员链路上配置**no cdp enable**，尤其是如果消息用于Cisco发现协议(CDP)数据包。
- 捆绑包的所有成员链路都应启用keepalive。

限制

以下是所有这些分布式功能的一般限制：

- T1和E1线路不能混合在捆绑中。
- 支持的最大差分延迟为30毫秒。
- 捆绑包中的所有线路必须位于同一端口适配器(PA)上。
- 不支持硬件压缩。
- VIP或FlexWAN CEF仅限IP;所有其它协议都发送到RSP。
- dMLP和dMLFR的传输端不支持分段。
- dLFI不支持许多较旧的排队机制。这些机制包括：虚拟模板接口上的公平队列虚拟模板接口上的随机检测自定义队列优先级队列
- 公平队列、随机检测(dWRED)和优先级队列可在具有模块化QoS CLI的流量策略中配置。
- 使用dLFIoFR或dLFIoATM时，每个MLP捆绑仅支持一条链路。如果使用dLFIoFR或dLFIoATM时，如果MLP捆绑中使用多个链路，则dLFI将自动禁用。在租用线路上使用dLFI时，可在MLP捆绑包中为多个链路配置dLFI。
- 使用dLFIoATM时，仅支持aal5snap和aal5mux。不支持封装al5nlpid和aal5ciscopp。
- 仅支持IP语音；dLFI功能不支持帧中继语音和ATM语音。
- 使用以下功能组合时，不应在多链路接口上配置压缩实时协议(CRTP)配置：启用LFI的多链路接口多链路捆绑包具有多个成员链路在多链路接口上启用具有优先级功能的QoS策略

使用dMLP和dLFI配置时，优先级数据包不传送MLP报头和序列号，并且MLP将在所有成员链路上分配优先级数据包。因此，CRTP压缩的数据包在接收路由器处可能无序到达。这禁止CRTP解压数据包报头并强制CRTP丢弃数据包。

建议

建议捆绑包中的成员链路具有相同的带宽。如果向捆绑包添加不等带宽链路，将导致大量数据包重新排序，从而降低整个捆绑包吞吐量。

建议将VIP2-50（带8 MB SRAM）或更高版本与这些分布式功能配合使用。

捆绑包、链路和内存要求的数量

请参阅Cisco 7500系列路由器的分布式多链路点对点协议。

7600 SIP线卡上的硬件和软件MLPPP或MLFR

MLP和MLFR可以基于软件或硬件。在基于硬件的MLP或MLFR中，Freedm提供多链路功能，MLP报头由Freedm芯片添加。在基于软件的MLP或MLFR中，SIP线卡CPU提供多链路功能（与VIP和FlexWAN实施类似）。

这些是运行基于硬件的MLP或MLFR的限制和条件。

- 每个线卡最多只能有336个捆绑包，每个安全状态评估(SPA)（自由）最多只能有168个捆绑包。
- 每个捆绑包最多只能有12个DS1/E1。
- 所有链路应属于同一SPA(Freedm)。
- 捆绑包中的所有链路应以相同的速度运行。
- TX分片大小可以是128、256或512。CLI配置的分片大小会映射到最近支持的分片大小。

```
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分片大小可以是1到9.6 KB。
- 不支持思科专有格式(MLFR)。

在硬件LFI中，如果捆绑中只有一条链路，如果是DS1/E1，则分段和交织将由Freedm完成。

show ppp multilink的**输出显**示硬件实施是否正在运行。

```
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.
```

如果多链路基于软件，则**show ppp multilink输**出的输出中中的多链路。

数据包的寿命

Rx数据路径

1. 驱动程序接收的数据包。
2. 检查封装：如下所示基本封装：在dMLP中，入口接口的封装类型为ET_PPP。在dMLFR中，入口接口的封装类型为ET_FRAME_RELAY。在dLFIoLL中，入口接口的封装类型为ET_PPP。在dLFIoFR中，入口接口的封装类型为ET_FRAME_RELAY。在dLFIoATM中，入口接口的封装类型为ET_ATM。在dDialer中，封装类型为ET_PPP。其他封装处理：对于ET_PPP，NLPID被嗅探出去。对于dMLP，NLPID是MULTILINK。对于dLFIoLL，需要考虑两点：VoIP数据包 — 这些数据包没有MLP报头，并且有指示IP的NLPID。数据包 — NLPID为MULTILINK。对于dDialer，数据包将没有MLP报头，并且具有指示IP的NLPID。**注意：**在本例中，您可以配置dCRTP（分布式压缩实时协议）。 如果是，在进一步处理之前解压报头。
3. 对于ET_FRAME_RELAY，如果接收数据包的链路配置为dMLFR，则数据包将处理为dMLFR
4. 对于dLFIoFR和dLFIoATM，封装类型分别为ET_FRAME_RELAY和ET_ATM;但其中有PPP报头。与dLFIoLL一样，PPP报头将指示数据包是语音数据包还是数据包。如果配置了dCRTP，则在进一步处理之前解压报头。语音数据包会立即交换。分片的数据包在交换之前必须重组。使用ET_PPP时，您可能会遇到PPP链路数据包；而使用ET_FRAME_RELAY时，您可能会遇到MLFR控制数据包。所有这些控制数据包都会传送到RP进行处理。
5. 根据上述解码，检查数据包所需的交换类型。链路类型将确定数据包是IP交换还是MPLS交换。然后，将所述分组分配给各自的交换功能。
6. 通过捆绑与分布式功能相结合，IP Turbo快速交换向量被盗。这是因为数据包在成员链路上接收；但是，必须对其进行处理，使其在捆绑包上接收。您还需要检查是否有控制数据包被传送到主机。主要在dMLFR中，有本地管理接口(LMI)数据包，它们不是*MLFR控*制数据包。对于这些，使用dLCI编号空间的不同部分。每当dLCI解码为落入此空间时，数据包就被传送到主机，因为它被识别为LMI数据包。VoIP数据包（在低延迟队列中排队）在不添加MLP报头的情况下才被切出。当接收到分段的数据包时，分布式功能可以接收和重组数据包。重组过程将在后面的章节中介绍。如果数据包必须进行标记交换，则会在dMLP中将其传递到标记交换例程。否则，如果它要进行IP交换，则会传递到IP交换例程。**注意：**所有非IP数据包都会传送到dMLFR中的主机。
7. **IP:**IP交换功能对所有数据包都是通用的。它主要做三件事：如果配置了任何功能，请对数据包执行必要处理。此外，当使用分布式拨号器时，当收到“相关数据包”时，在此处执行空闲计时器更新。有关空闲**计时器配置参数的详细信息**，请**参阅拨号器空闲超时（接口）、拨号器快速空闲（接口）和配置拨号器配置文件**。在75xx路由器上，邻接关系将指示的tx_acc_ptr。如果出口接口是虚拟访问接口，则`tx_acc_ptr`为`NULL`。在这种情况下，请修复封装并从fib `hwidb` tx_acc`_ptr`。在dLFIoFR和dLFIoATM中，必须进行此查找和封装修复。在dLFIoLL中，链路被视为多链路捆绑的一部分。**注意：**此处调整了数据包的TTL，并检查了IP分段。所有mci_status设`RXTYPE_DODIP`。
8. 在做出交换决策后，数据包即可从接口发出。检查接口以确定其是否支持本地交换。如果有，则直接通过fastsend发送。否则，会尝试路由缓存交换它。请注意，如果为接口配置了QoS，则本地交换矢量会被QoS盗用。HQF将对数据包进行排队并进一步处理该数据包，然后才将其最终从接口发送出去。dLFI就是这样。对于dLFI，会设置分段和交织。QoS处理分段例程的调用，并将分段数据包与将在优先级队列中排队的语音数据包（如果配置了LLQ）交织。这可确保VoIP数据包不会因通过链路传输大量数据包所需的延迟而受到影响。

Tx数据路径

vip`_dtq_consumer`取数据包并获取接口编号，从中获取`idb`。与idb对应的fastsend例程:

i)快速发送

1. 在dMFR中，fr_info结构从表中检索，该表将if_index与fr_info匹配。控制数据包刚刚发送出去

。帧报头将提供dLCI，这将帮助您确定这是LMI数据包还是数据包。帧头中的dlci字段被dmfr序列号覆盖。LMI和数据包使用单独的序列号。**注：单**独的序列号用于单独的dLCI。
2. 在dMLP中，控制数据包的优先级设置为高。如果配置了dCRTP，则对报头进行压缩。VIP MLP报头（包括排序信息）会添加并从成员链接发出。
3. 在dLFI中，HQF拦截要通过接口发送的数据包。如果语音数据包是语音数据包，则语音数据包被置于优先级队列（如果配置了LLQ）中，并在不使用MLP封装的情况下从接口发出。对于数据包，它调用dLFI分段代码，该代码将分段返回到QoS代码，然后与优先级流量交织，以满足语音流量的延迟要求。此外，如果配置了dCRTP，则仅压缩语音数据包的报头。数据包报头保持原样。
4. 在dDialer中，对数据包进行分类，以在发送数据包之前重置输出链路的空闲计时器。如果多个链路绑定到同一拨号器，则在选择输出链路后执行此操作。拨号器数据包中未添加报头。因此，拨号器接口不支持数据包的排序和重组。

注意：在具有多个链路的dMLP、dDialer、dMLFR和dLFI中，转发流量所依赖的物理链路取决于链路拥塞。如果链路拥塞，请移至下一条链路，依此类推。(dMLFR、没有QoS的dMLP和dDialer功能还根据链路上的字节数选择链路。如果当前链路已经发送了其字节配额，它会选择下一条链路（轮询）。此配额由链路的frag_bytes决定。对于拨号器成员接口，frag_bytes设置为接口带宽的默认值。）

注意：在出口VIP接口上的HQF配置中，HQF会窃取dtq_consumer矢量。到出口VIP的数据包DMA将首先通过HQF检查。如果在出口接口上配置了QoS，则HQF会在数据包从接口快速发送出去之前开始处理该数据包。

重组

普通dDialer接口不支持重组和排序。要在拨号器接口上启用此功能，必须配置拨号器接口上的MLP。如果执行此操作，则Rx和Tx路径与dMLP路径相同。收到数据包时，会根据预期序列号检查序列号。

- 如果序列号匹配：如果数据包是未分段的数据包，则无需重组。继续执行进一步的交换步骤。如果数据包是分段，则检查开始和结束位，并在收到分段时构建数据包。
- 如果序列号不匹配：如果序列号在序列号的预期窗口内，则将其放入排序的“未分配分段列表”中。 之后，当未收到预期序列号时，会检查此列表，以防数据包存储在此处。如果序列号不在窗口内，请将其丢弃并报告“已接收丢失的片段”。 如果稍后在等待此数据包时超时，接收方将重新同步，并从收到的下一个数据包开始。

在所有这些情况下，都会从此接口发送正确排序的数据包流。如果收到分段，则形成完整的数据包，然后发送出去。

配置、检验和调试分布式功能

本节介绍可用于验证和调试每个**分布式功**能的**show**和**debug**命令。

配置和检验dMFR

MFR配置示例

```
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接口与另一个FR接口类似，因此支持大多数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
```

验证RP上的MFR捆绑包状态

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
```

这表示两个接口已正确添加，并且一个接口尚未协商MLFR LIP消息。

要获取有关MFR捆绑包和成员链接的详细信息，请发出以下命令：

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调试命令

这些调试对于解决链路未添加到捆绑包的问题非常有用。

debug frame-relay multilink control

注：当未指定特定MFR接口或串行接口时，这将启用所有MFR链路的调试。如果路由器有大量MFR链路，这会非常困难。

要调试在RP处收到的MFR数据包以及调试MFR控制活动，此调试非常有用：

debug frame-relay multilink

注意：在流量较大的情况下，这会使CPU不堪重负。

验证LC上的dMLFR捆绑包状态

show frame-relay multilink

注意：目前，LC上不提供此功能，但很快会添加。在此之前，请使用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
```

配置和检验dMLP/dLFIoLL

多链路PPP配置

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

串行接口下的配置示例：

```
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
!
```

注意：ppp **chap hostname M1命**令并不表示CHAP身份验证已启用。此命令中的**字符串M1**充当端点标识符，仅在同一两台路由器之间将有多个多链路捆绑时才需要。在这种情况下，属于捆绑的所有链路都应具有相同的端点标识符，而属于不同捆绑的两个链路不应具有相同的端点标识符。

可选配置参数

[no] ppp multilink interleave

这可在多链路捆绑上启用交织。这与模块化QoS CLI配合使用。高优先级数据包将在不添加MLP序列和报头的情况下传输，而其他数据包将分段并随MLP序列和报头一起传输。

注意：当使用多条链路启用交织时，高优先级流量可能会重新排序。启用或禁用交织时，需要重置捆绑才能在线卡中激活它。

ppp multilink mrru local *value*

这指定多链路上的最大接收单元；多链路接口将接受大小达此大小的数据包。此处的大小不包括MLP报头。

ppp multilink mrru remote *value*

这指定远程端应支持的最小MRRU。如果远程终端MRRU小于此值，则捆绑协商将失败。

ppp multilink fragment delay *seconds*

这指定由数据分段导致的允许延迟（毫秒）。换句话说，延迟值用于计算允许的最大分段大小。分布式实施与Cisco IOS实施在以下方面不同：

1. 除非启用交织，否则不执行分段。
2. 对于带宽不同的链路，选择的分段大小取决于带宽最小的接口。

ppp multilink fragment disable

此命令不会在分布式实施中添加任何功能。仅当启用交织时才会发生分段；并且，启用交织后，将忽略ppp multilink fragment disable命令。

验证RP上的dMLP捆绑包状态

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. 当捆绑包处于分布式模式时，show ppp multilink的输出中会显示以下信息：如果不是，则不会因某种原因分发捆绑包。
2. 在线**卡上配置并启**用ppp多链路交织时，show ppp multilink**输出包**括dLFI统:`Fragemented` — 表示发送和接收的分段计数。`Unfragemented` — 表示在未分段的情况下传输或接收的数据包的计数。`Reassembled` — 指示重组的完整数据包数。未启用交织时，输出如下所示：

```
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
```

上例中的分片大小为760字节。

验证LC上的dMLP捆绑状态

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时，序列号按dLCI进行维护，捆绑包中的序列号用于LMI dLCI。

字段	描述
dmlp_ipc_config_count	LC为多链路或MLFR配置接收的IPC消息数
dmlp_bundle_count	LC中MLP和MLFR包的数量
dmlp_ipc_fault_count	导致LC失败的配置消息数。应为0;如果它不是零，则可能有问题。
	指示标记交换中使用的idb to tag_optimum_fs和idb to ip2tag_optimum_fs矢量。
board_encap	指示用于添加2个字节的板封装的board_encap矢量（如果7500平台中有信道化链路）。如果链路包含非信道化接口，则应为NULL。
max_particles	可容纳在重组缓冲器中的最大粒子数
mrru	在不考虑MLP封装的情况下接受的最大数据包大小。不适用于MLFR接口。
seq_window_size	序列号的最大窗口大小
working_pak	指示当前在重组中的pak。NULL，如果无。
working_pak_cache	指向用于重组的静态pak的指针。当捆绑包接收到第一个未完成数据包时，会分配该数据包。
una_frag_list	重组队列中的第一个条目。如果条目不为NULL且不更改，则表明计时器未运行软件问题。
una_frag_end	重组队列中的最后一个条目
rcved_end_bit	表示捆绑包已接收结束位，因此它正在寻找开始位。
is_lost_frag	如果碎片声明丢失，则为true。当收到具有预期序列的分片时，将清除此项。
resync_count	指示接收方与发射器不同步并必须通过从最后接收的序列片段开始重新同步的次数。

	表示已发生重组超时，且正在从重组队列处理数据包。
timer_start	重组计时器启动的次数
timer_running	指示重组计时器是否正在运行。
timer_count	指示重组计时器已过期的次数。
next_xmit_link	传输下一个数据包的链路
	位字段，表示存在的成员。
	字段不在所有分支中使用。指示哪些成员链路未拥塞。
dmlp_orig_pak_to_host	用于将数据包传送到RP的矢量。
dmlp_orig_fastsend	在MLP或MLFR修改驱动程序的fastsend之前，原始驱动程序的fastsend。
	丢失的分段数（接收方未接收这些分段）。 当向主机发送更新时，系统会定期清除此项。
	按预期顺序接收的分段数。当向主机发送更新时，系统会定期清除此项。
	由于无法生成完整数据包而丢弃的分段数
	收到的据信已丢失的分段数。这表示链路间延迟大于dMLP重组超时30毫秒。

配置和检验dLFIoFR和dLFIoATM

```
class-map voip
 match ip precedence 3

policy-map llq
 class voip
  priority

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


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

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

验证RP上的dLFIoFR/ATM捆绑状态

```
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：仅当在虚拟模板下配**置ppp多链**路交织时，捆绑才会分发；如果没有此命令，则不会分发捆绑包。

验证LC上的dLFIoFR/ATM捆绑状态

要验证dLFI确实在LC上正常工作，请发出以下命令：

```
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 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
```

应该有优先级层和WFQ层。分段将在WFQ枝叶层块完成。

配置和检验dDDR

当您在全局配置上启用**ip cef distributed**和在拨号器接口上**分配的ip route-cache时，**分布式DDR会激活。

```
!--- 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
```

分布式DDR没有其他特殊配置。进一步配置遵循正常DDR配置。

验证分布式按需拨号路由

```
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: 0x807FFFFF 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
```

拨号告诉我们使用的拨号器类型。ISDN示传统拨号程序配置，而PROFILE示拨号程序配置。拨号指示拨号器的当前状态。未连接拨号器接口的状态将。每当量时，Idle计时器就会重置。如果此计时器过期，接口将立即断开连接。器是可配置参数。有关详细信息，请参阅使用拨号程序配置文件配置点对点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
```

显示的变量与dMLP的变量相同。

调试dMLP和dDDR

RP上可用的调试

dDDR

debug dialer [events | packets | forwarding | map]

发出此命令可调试控制路径功能，如呼叫设置等。有关详细信息，请参阅debug dialer events。

debug ip cef dialer

发出此命令以调试与CEF相关的拨号器事件。有关详细信息，请参阅拨号器CEF。

LC上可用的调试

dMLP

控制路径调试：**debug multilink event**

数据路径调试：**debug multilink fragments**

数据路径和控制路径错误调试：**debug multilink error**

在SIP线卡上调试dMLP

根据CI转储数据包：数据包和控制数据包可以根据控制ci和序列ci转储到线卡上。

test hw-module subslot_*num* dump ci *CI-NUM [rx|tx] num_packets_to_dump*

CI可以通过以下方式获得：

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

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

SPA 6/0 base address 0xB8000000 efc 1

Interface Serial6/0/0:0 is administratively down
Type 0xD Map 0x7FFFFFFF, Subrate 0xFF, mapped 0x1, maxmtu 0x5DC
Mtu 1500, max_buffer_size 1524, max_pak_size 1608 enc 84
ROM rev: 0, FW OS rev: 0x00000000 Firmware rev: 0x00000000
  idb=0x42663A30, pa=0x427BF6E0, vip_fci_type=0, port_per_spa=0
  SPA port type is set
  Host SPI4 in sync
  SPA=0x427BF6E0 status=00010407, host=00000101, fpga=0x427EDF98
  cmd_head=113, cmd_tail=113, ev_head=184, ev_tail=184
  ev_dropped=0, cmd_dropped=0
!--- Start Link Record Information.  tag 0, id 0, anyphy 0, anyphy_flags 3, state 0
 crc 0, idle 0, subrate 0, invert 0, priority 0
 encap hdlc
 corrupt_ci 65535, transparent_ci 1
!--- End Link Record Information. Interface Serial6/0/0:0 is administratively down Channel
Stats: in_throttle=0, throttled=0, unthrottled=0, started=1 rx_packets=0, rx_bytes=0,
rx_frame_aborts=0, rx_crc_errors=0 rx_giants=0, rx_non_aligned_frames=0, rx_runts=0,
rx_overruns=0 tx_packets=0, tx_bytes=0, tx_frame_aborts=0 is_congested=0, mapped=1, is_isdn_d=0,
tx_limited=1 fast_if_number=15, fastsend=0x403339E4 map=0x7FFFFFFF, turbo_vector_name=Copperhead
to Draco switching lc_ip_turbo_fs=403A9EEC, lc_ip_mdfs=403A9EEC
```

对于CT3，必须获取vc num，该vc num可以从show interface serial **CT3_*interface_name的输出中获得*。**

现在，可以从SPA控制台获取CI信息。首先，使用spa_redirect rp ct3_freedm336命令将SPA控制台命**令的输出重定向到RP。**

spa_ct**3_test freedm show linkrec vc命令显**示必要的CI信息。

dMFR

控制路径调试：**debug dmfr event**

数据路径调试：**debug dmfr packets**

数据路径和控制路径错误调试：**debug dmfr error**

根据CI转储数据包：参见dMLP。

dLFI

控制路径调试：**debug dlfi event**

数据路径调试：**debug dlfi fragments**

数据路径和控制路径错误调试：**debug dlfi error**

dDDR

没有特殊的调试命令；应使用dMLP调试。

在dLFIoLL的情况下，可能需要同时使用dMLP和dLFI调试。这些调试不是有条件的，因此将触发所有捆绑包。

常见问题

1. **什么是dMLP?**dMLP是分布式多链路PPP的简称(如RFC1990中所述)。 Cisco 7500系列和7600系列等分布式平台支持此功能。dMLP允许您将T1/E1线路（在Cisco 7500系列路由器的VIP中）或7600系列路由器的FlexWAN中)合并到具有多条T1/E1线路的合并带宽的捆绑包中。这样，客户无需购买T3/E3线路即可增加T1/E1以外的带宽。
2. **dMLP中的"分布式"是什么？**术语"分布式"表示数据包交换由VIP而不是RSP完成。为什么？RSP交换功能相当有限，需要做的工作也比这重要得多。能够交换数据包的VIP从RSP卸载此活动。基于RSP的Cisco IOS仍然管理链路。捆绑包的创建和拆卸由RSP完成。此外，RSP仍然会处理PPP控制平面，包括处理所有PPP控制数据包（LCP、身份验证和NCP）。但是，一旦建立捆绑，MLP数据包的处理将转交给VIP，由板载CPU进行交换。dMLP引擎（在VIP上）处理所有MLP过程，包括分段、交织、封装、多个链路之间的负载均衡以及入站分段的排序和重组。VIP在7500系统中完成的功能由Flexwan/Enhanced-FlexWAN在基于7600的系统中完成。
3. **如何确定捆绑包是否已分发？**在路由器控制台**发出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)
    Se11/1/0/27:0, since 00:22:46, no frags rcvd
    Se11/1/0/25:0, since 00:22:46, no frags rcvd
!--- Output suppressed.
```

4. **如果我升级到RSP16或SUP720，我的dMLP性能会更好吗？**否。dMLP（或任何分布式功能）的交换性能取决于相关VIP或FlexWAN。例如，VIP6-80的性能将优于VIP2-50。
5. **我可以使用哪些PA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（仅7500）
6. **一个捆绑包中可以配置多少条链路？**这个答案有很多方面。主要瓶颈是线卡(VIP/FlexWAN/Enhanced-FlexWAN2)的CPU功率。 硬限制是每个捆绑包56条链路，但是，由于CPU电源或缓冲区有限，您很多时候无法配置这些链路（并且有如此多的流量交换）。这些数字基于此指南（基于VIP/FlexWAN/Enahnced-FlexWAN2上的CPU和内存）：VIP2-50（带4MB SRAM）最大T1 = 12VIP2-50（带8MB SRAM）最大T1 = 16VIP4-80最大T1 = 40VIP6-80最大T1 = 40FlexWAN最大T1 =将很快更新Enhanced-FlexWAN最大E1 =每个托架21个E1（每线卡聚合42个E1）
7. **如果我为3个捆绑包配置3个T1，或为1个捆绑包配置9个T1,*性能*是否会发生变化？**性能没有变化，经实验室测试证明。但是，由于单个捆绑包中有大量T1（例如单个捆绑包中有24或28个T1），因此存在缓冲区耗尽问题。它强烈建议您在单个捆绑包中不要超过8个成员链路(T1/E1)。
8. **如何确定捆绑包的带宽？**不配置捆绑包的带宽。它是所有成员链路的聚合带宽。如果捆绑包中有4个T1，则捆绑包的带宽为6.144Mbps。

9. **哪个更好？CEF — 负载均衡还是dMLP?**这个问题没有简单的答案。你的需求决定了哪个更好。**MLP的优点：**CEF负载均衡仅适用于IP流量。MLP平衡通过捆绑发送的所有流量。MLP维护数据包的顺序。IP本身容忍重新排序，因此这对您来说可能无关紧要；事实上，维护测序过程所涉及的额外成本可能是避免MLP的原因。IP适用于可能无序传送数据报的网络，而任何使用IP的网络都应能够处理重新排序。然而，尽管存在这一事实，但现实情况是，重新排序仍可能构成一个真正的问题。MLP提供到对等系统的单个逻辑连接。多链路捆绑包支持QoS。MLP提供动态带宽功能，因为用户可以根据当前需求添加或删除成员链路。MLP可以捆绑更多链路，而CEF负载均衡限制为6个并行IP路径。每个流的CEF负载平衡将任何给定流的最大带宽限制为一个T1。例如，使用语音网关的客户可以拥有许多具有相同源和目标的呼叫，因此只使用一条路径。**MLP的缺点：**MLP会为每个数据包或帧增加额外开销MLP占用CPU;dMLP占用线卡CPU。
10. **如何在两台路由器之间配置多个捆绑包？**多链路根据对等体的名称和端点标识符确定链路将加入的捆绑。要在两个系统之间创建多个不同的捆绑包，标准方法是强制某些链路以不同方式标识自身。推荐的方法是使用ppp chap hostname **name命令*进***行。
11. **我是否可以有来自不同PA的成员链接？**否。如果要运行dMLP，则不支持它。但是，如果从不同的PA添加成员链路，则该控件将再提供给RSP及其不是dMLP。MLP仍在运行，但dMLP的优势已不复存在。
12. **能否将两个托架的成员链路混合使用？**否。如果要运行dMLP，则不支持它。但是，如果成员链路是从不同PA添加的，则该控件会被赋予RSP，而不再是dMLP。MLP仍在运行，但dMLP的优势已不复存在。
13. **我是否可以在不同的VIP或FlexWAN之间拥有成员链路？**否。如果要运行dMLP，则不支持它。但是，如果从不同的PA添加成员链路，则该控件将再提供给RSP及其不是dMLP。MLP仍在运行，但dMLP的优势已不复存在。
14. **我是否可以通过单个PA的不同端口建立成员链路？**（例如，从PA-MC-2T3+的每个CT3端口发出一个成员链路。）Yes.只要它来自同一个私人助理，就没有问题。
15. **能否捆绑T3或E3端口？**否。对于7500/VIP、7600/FlexWAN和7600/FlexWAN2，仅允许使用dMLP的DS0、n*DS0、T1和E1速度。**注意：**仅以T1/E1或子速率T1/E1速度配置的成员链路支持分布式MLPPP。信道化STM-1/T3/T1接口还支持dMLPPP，速度为T1/E1或子速率T1/E1。以净通道T3/E3或更高接口速度配置的成员链路不支持分布式MLPPP。
16. **什么是"重排"片段？**如果收到的分段或数据包与预期序列号不匹配，则重计数器将递增。对于不同的数据包大小，这是必然的。对于固定大小的数据包，这也可能是因为PA驱动程序处理在一条链路上接收的数据包，而不是按循环方式（在dMLP中传输数据包时执行）。 重新排序并不意味着丢包。
17. **什么是"丢失"片段？**每当接收到无序片段或数据包时，您发现所有链路上都接收到无序片段或数据包时，丢失的分片器就会递增。另一种情况是，无序片段被存储在列表中，并且达到限制（根据VIP上的SRAM和为捆绑分配的任何内容确定）时，丢失片段计数器递增，并且列表中的下一个序列号被处理。
18. **dMLP如何检测丢失的分段？**序列号:如果您正在等待序列号为N的片段到达，并且所有链路都收到序列号大于N的片段，则您知道片段N必须丢失，因为它不可能合法到达同一链路上编号较高的片段后面。超时：如果你坐得太久，等待碎片，你最终会宣布碎片丢失，然后继续。重组缓冲区溢出：如果您正在等待片段N到达，同时其他片段（序列号高于N）正在到达某些链路，则您必须将这些片段保留在重组缓冲区中，直到片段N显示。缓冲量是有限的。如果缓冲区溢出，则再次声明片段N为丢失，并使用缓冲区中的任何内容恢复处理。
19. **什么是"丢失接收"？**丢失接收的分段或数据包可能有两个原因：如果收到的分段或数据包超出预期的序列范围窗口，则通过将其标记为已丢失的已接收数据包来丢弃该数据包。如果收到的分段或数据包在预期序列范围窗口内，但您无法分配数据包报头来重新生成此数据包，则数据包将被丢弃并标记为已丢失。
20. **dMLP是否支持加密？**不能。

21. **我们是否支持PFC报头压缩？**不，不在分布式路径中。不建议远端路由器配置PFC报头压缩，因为如果收到压缩报头帧或数据包，我们会回退到非分布式模式。如果要继续运行dMLP，必须在两端禁用PFC报头压缩。
22. **dMLP是否支持软件压缩？**否，因为软件压缩在分布式路径中不起作用。
23. **传输端是否支持分段？**不是香草dMLP。使用Vanilla dMLP接收分片没有问题，但在传输端，不会发生分片。当在dMLP接口上配置**ppp multilink interleave时**，支持传输端分段。
24. **我们是否能ping通MLP捆绑包的成员链路？**否，您无法在成员链路上配置IP地址。
25. **链路MTU和MLP分段大小是否存在依赖关系？**否。MTU大小与MLP分片大小无关，但MLP分片与任何其他帧一样不能超过串行链路的MTU大小是明显的限制。
26. **能否在一对路由器之间配置两个MLP捆绑包？**是的，这是可能的。但是，这可能导致负载均衡受损。它可用于测试平台，仅使用两台路由器模拟两台以上的路由器，但它没有任何明显的实际价值。必须将指向公用对等体的*所有*链路放入同一捆绑包中。根据定义，捆绑包是指向特定对等体的一组链路。“对等体”由它在LCP和身份验证阶段提供的用户名和端点标识符值标识。如果您尝试在两个路由器之间创建多个捆绑包，则意味着您尝试将每个路由器伪装成与其对应路由器不止一个对等体。他们必须恰当地自我标识。
27. **成员链路能否使用不同的排队算法？**与捆绑相关的所有排队机制都需要在捆绑级别应用，而不是在成员链路级别应用。但是，配置队列算法不应影响数据包从捆绑包中的交换方式。
28. **当在Cisco 7500上启用dMLP时，为什么将tx-quque-limit设置为26作为多链路捆绑的成员链路的默认值？**对于带宽T1/E1的任何串行接口，tx-queue-limit大约为4或5。当您将T1/E1捆绑在多链路中时，捆绑的带宽将增加。由于交换将基于MLP接口的带宽进行，因此您需要增加成员链路的tx-queue-limit。只有一个成员链路（称为主链路）用于交换，因此需要增加其tx-queue-limit。此外，此值是经过测试后选择的经验值，然后调整到此值。通常，部署的捆绑包中的T1/E1链路不超过4到6条。值26可以完美满足6到8条T1/E1链路，因此选择了此值。
29. **dMLP实施中的差分延迟及其值是什么？**dMLP支持30毫秒的差分延迟。这意味着，如果分段在某个时间T接收，并且其顺序混乱（预期序列号为100，但我们收到101）。 如果序列号100在T+30毫秒之前未收到，则100将被声明为丢失，如果您可以从101开始处理，则会执行此操作。如果不能以101开头（如果是中间片段），则会查找具有起始片段（例如，104）的片段，然后从此开始。
30. **当数据包在IP级别与7500上的多链路分段时，会发生什么情况？**如果数据包在IP级别进行分段，则它们在中间跳处无需重组即可传输，但在目的路由器处重组。
31. **当数据包在7500的MLP级别分段时，会发生什么情况？**如果数据包在MLP级别分段，并且重组后的数据包大于MRRU，则数据包在多链路上被丢弃。仅在dMLP上支持传输端分段（仅使用dLFI）。只有当packet_size大于frag_size且小于MRRU时，数据包才会在MLP级别分片。如果发送的数据包超过MRRU，并且未在IP级别分段，则另一端会丢弃在MLP级别未分段的所有数据包大小，因为数据包超过MRRU。
32. **如何计算MRRU?**MRRU根据以下首选项计算：对于进入的新成员链路，根据在成员链路上配置的MRRU，在LCP级别再次协商MRRU。在链路接口上使用ppp multilink mrru interface命令**配置的**值。如果未配置，则从父接口上ppp multilink mrru命令的**配置**继承的值。如果两个值都存在，则链路接口值优先。默认MRRU为1524。

调试增强功能

这些增强功能将在将来实施。规划尚未完成。

- 在LC**上启用debug frame-relay multilink**命令。
- 增强每个接口的当前调试CLI和指定数量的数据包。
- 对于dDDR，QoS功能尚不受支持。这只能通过适当的业务案例解决。

相关信息

- 拨号器CEF
- 使用 Dialer Profile 配置对等 DDR
- MPLS — 多链路PPP支持
- 适用于Cisco 7500系列路由器的分布式多链路点对点协议
- 分布式多链路帧中继(FRF.16)
- 租用线路上的分布式链路分段和交织
- 带有服务质量控制（LLQ/IP RTP 优先级、LFI、cRTP）的 VoIP-over-PPP
- 技术说明故障排除 — Cisco 7500系列路由器
- 路由器产品支持页 — Cisco Systems
- 技术支持和文档 - Cisco Systems