在Cisco 75xx和76xx路由器上的被分配功能的配置和验证

目录

简介

本文帮助您了解，配置和验证这些功能：

- 分布式多链路点对点协议(dmlp)
- Distributed Link Fragmentation and Interleaving (LFI)
- 在租用的线路(dLFIoLL)的分布式LFI
- 在帧中继(dLFIoFR)的分布式LFI

- 在ATM (dLFIoATM)的分布式LFI
- 在分布式MLP (dmlp)和dLFIoLL之间的区别
- 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

本文读者应该有分布式功能知识思科的7500/7600/6500。

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，此表列出：

每个分组和平台的分布式功能支持

功能	端口适配器(PA)[1]支持	7500平台		7600平台	
		主要Cisco IOS软件软件版本	Cisco IOS版本(过渡技术)	主要Cisco IOS软件软件版本	Cisco IOS软件版本(过渡技术)
dmlp	Chan-PA PA-4T+ PA-8T	12.0T 12.0S 12.1 12.1T 12.2 12.2T 12.3 12.3T 12.2S[2] 12.1E	12.0(3)T及以后12.0(9)S和以后	12.2SX[2] 12.1E	
dLFIoLL	Chan-PA PA-4T+ PA-8T	12.2T 12.3 12.3T 12.0S	12.2(8)T及以后 12.0(24)S和以后	12.2SX	12.2(17)SXB和以后
dLFIoFR	Chan-PA PA-4T+ PA-8T	12.2T 12.3 12.3T	12.2(4)T3和以后	12.2SX	12.2(17)SXB和以后

dLFIoATM	PA-A3 PA-A6	12.2T 12.3 12.3T	12.2(4)T3和以后	12.2SX	12.2(17)SXB和以后
dMLFR	Chan-PA PA-4T+ PA-8T	12.0S 12.3T	12.0(24)S及以后 12.3(4)T和以后	12.2SX	12.2(17)SXB和以后
在dmlp的QoS	Chan-PA PA-4T+ PA-8T	12.0S 12.2T 12.3 12.3T	12.0(24)S及以后 12.2(8)T和以后	12.2SX	12.2(17)SXB和以后
在dmlp MPLS的MPLS在dLFIoLL	Chan-PA PA-4T+ PA-8T	12.2T 12.3	12.2(15)T11及以后 12.3(5a)及以后	12.2SX	12.2(17)SXB和以后
分布式DDR	PA-MC-xT1 PA-MC-xE1 PA-MC-xTE1 PA-MCX-xTE1	12.3T	12.3(7)T和以后		

注意： 注意此信息：

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支持在7500和7600平台的这些功能。

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

有关文档规则的详细信息，请参阅 Cisco 技术提示规则。

分布式功能

这些功能在本文解释：

- 分布式MLP
- 分布式LFI
- 在租用的线路的分布式LFI
- 在帧中继的分布式LFI
- 在ATM的分布式LFI
- 在dmlp和dLFIoLL之间的区别
- 分布式MLFR
- 分布式拨号程序
- 支持分布式功能的平台和线路卡

分布式MLPPP

分布式多链路点对点协议(dmlp)功能允许您结合在一线卡(VIP的全双工或小部分T1/E1线路，FlexWan)在一个Cisco 7500或7600系列路由器到有复合带宽多条链路的套件。您使用一个分布式MLP套件执行此。用户能选择套件数量在路由器的和链路数量每个套件。这允许用户增加带宽在那的网络链路单个T1/E1线路之外，不用需要采购T3线路。在非dMLP，所有数据包由路由处理器(RP)交换;因此，此实施影响RP的性能，与运行MLP的仅一些条T1/E1线路的高CPU利用率。使用dmlp，在路由器可以被处理套件的总数增加，因为数据路径由线路卡CPU和内存处理并且限制。dmlp允许您从DS0 (64 Kbps)开始向前捆绑小部分T1/E1。

分布式LFI

dlfi功能支持实时数据流(例如语音)和非实时流量传输(例如数据)在低速度帧中继和ATM虚拟电路(VC)和在没有导致额外延迟的租用的线路实时数据流。

此功能实现使用多链路PPP (MLP)在帧中继、ATM和租用的线路。功能分段大型数据包到更加小的片段顺序，启用延迟敏感实时数据包，并且共享同样的非实时数据包连接。片段用实时数据包然后插入。在链路的接收端，片段被重新组装，并且数据包重建。

dlfi功能经常是有用的在通过低延迟排队(LLQ)发送实时数据流的网络(例如语音)，但是有带宽问题。这延迟实时数据流由于大，较不时间敏感的数据包传输。您在这些网络能使用dlfi功能，拆卸大数据包到多个网段。实时数据流数据包可以然后被发送在数据包的这些分段之间。在此方案中，当等待低优先级数据数据包穿程网络时，实时数据流不体验较的延迟。数据包被重新组装在链路的接收端，因此数据原封传送。

链路的分段大小根据在多链路捆绑的分段延时计算，配置用**ppp multilink fragment-delay n命令**，where:

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

此分段大小代表仅IP有效载荷。它不包含封装字节(分段大小=权重–封装字节)。期限“权重”和“分段大小”是如在输出**show ppp multilink命令中看到**在RP。如果分段延时没有配置，默认分段大小为最大片段延迟30计算。

注意： 使用变化的带宽链路，选定的分段大小根据与最少带宽的链路。

在租用的线路的分布式LFI

dLFIoLL功能对租用的线路扩大Distributed Link Fragmentation and Interleaving功能。分布式LFI用**ppp multilink interleave命令**配置在多链路组接口。是可行的您使用在多链路接口的分布式LFI与带

宽少于768 Kbps。这是因为1500个字节信息包的串行延迟带宽了不起的比768 Kbps的在可接受延迟限额内，并且不需要被分段。

在帧中继的分布式LFI

dLFIoFR功能是多链路PPP的分机在帧中继(MLPoFR)功能的。MLP使用分段。此功能类似于FRF.12，支持分段，并且能通过低延时队列插入高优先级数据包。

ppp multilink interleave命令在虚拟模板要求启用在相关的虚拟访问接口的交叉。除启用在serial interfaces的分布式CEF交换之外，此命令是一个前提对于工作分布式的LFI。

注意： 除非使用帧中继对ATM互联，推荐您使用FRF.12而不是dLFIoFR，因为带宽利用率是好与FRF.12

在ATM的分布式LFI

dLFIoATM功能是多链路PPP的分机在ATM (MLPoATM)功能的。MLP使用分段。

ppp multilink interleave命令在虚拟模板要求启用在相关的虚拟访问接口的交叉。除启用在serial interfaces的分布式CEF交换之外，此命令是一个前提对于工作分布式的LFI。

使用dLFIoATM，重要的是非常您选择做数据包适合ATM信元的分段大小，在这种情况下他们的ATM信元不导致多余的填充符。例如，如果选定分段大小是124个字节，这意味着124个字节IP有效载荷终于将去作为124个+ 10个(MLP报头) + 8个(SNAP信头) = 142个字节。请注意第一个片段将出去与124个+ 10个+ 2个(首先请分段PID报头大小) + 8个= 144个字节。这意味着此数据包将使用三个ATM信元转接有效负载，并且，因此，使用最高效地被包装的信元。

在dmlp和dLFIoLL之间的区别

dmlp不支持在传输端的分段，而dLFIoLL。

注意： 交叉和分段与超过一条链路一起使用在多链路捆绑语音流量不保证通过在套件的多条链路接收的语音流量按顺序将接收。语音正确排序被处理在上层。

分布式MLFR

分布式MLFR功能引入根据帧中继论坛多链路帧中继UNI/NNI实施协议的功能(FRF.16)对已启用线卡Cisco 7500和7600系列路由器。因为允许将聚集的多个串行链路到一个套件带宽，分布式MLFR功能提供一有效方法增加特定应用程序的带宽。用户网络接口(UNI)和网络对网络接口(NNI)支持MLFR在帧中继网络。

套件由多个串行链路做成，呼叫捆绑链路。在套件内的每个捆绑链路对应于物理接口。捆绑链路是隐身对帧中继数据链路层，因此帧中继功能在这些接口不可能配置。您要运用到这些链路的正常帧中继功能在捆绑接口必须配置。捆绑链路是可视对对等设备。

分布式DDR

分布式DDR功能允许在拨号接口的分布式交换。没有此功能，必须踢所有拨入流量到交换的主机。使用它，仅控制数据包被发送至RP，而交换决定完成在VIP，在连接被建立了后。

传统拨号程序配置和Dialer Profile配置仅支持与PPP封装。也支持在拨号接口的MLP。QoS不支持与在拨号接口的分布式交换。

分布式功能前提条件和限制

先决条件

这些是一般前提对于所有这些分布式功能：

- 必须启用Distributed Cisco Express Forwarding (DCEF)交换全局。
- 在成员serial interfaces必须启用DCEF交换，是MLP套件的一部分。
- 在dLFIoFR和dLFIoATM接口物理链路必须启用DCEF交换。
- 交错存取配置要求分配LFIoFR和LFIoATM。
- 配置在虚拟模板接口的所需的带宽dLFIoFR和dLFIoATM接口的。
- 当PPP调试在RP时启用，您也许观察MLP 在路由交换机处理器(RSP)的消息。由于此消息是混乱的，并且不需要—，特别是如果消息是为思科设备发现协议(CDP)数据包—您必须配置在套件成员链接的**no cdp enable**。
- 所有套件的成员链接应该有启用的Keepalive。

限制

这些是所有的一般限制这些分布式功能：

- T1 & E1线路在套件不可能被混合。
- 最大支持的差分延迟是30毫秒。
- 在套件的所有线路在相同端口适配器(PA)必须驻留。
- 不支持硬件压缩。
- VIP或FlexWan CEF对仅IP被限制;其他协议发送对RSP。
- dmlp和dMLFR的传输端不支持分段。
- dlfi不支持许多更旧的排队机制。这些机制包括：在虚拟模板接口的公平排队在虚拟模板接口的随机检测自定义队列优先级队列
- 公平排队、随机的检测(dWRED)和优先级队列在与模块化QoS CLI的一个数据流策略可以配置。
- 仅，当您使用dLFIoFR或dLFIoATM时，支持每个MLP套件一条链路。如果超过一条链路用于MLP套件，当时曾经dLFIoFR或dLFIoATM， dlfi自动地禁用。当曾经在租用的线路时的dlfi，超过一条链路可以配置与在MLP套件的dlfi。
- 使用dLFIoATM，支持仅AAL5SNAP和aal5mux。不支持封装aal5nlpid和aal5ciscopp。
- 支持仅基于IP的语音;dlfi功能不支持帧中继语音和Voice over ATM。
- 当您使用此功能组合时，在多链路接口不应该配置压缩实时协议(CRTP)配置：与启用的LFI的多链路接口多链路捆绑有超过一成员链接与优先级功能的QoS策略在多链路接口启用

使用dmlp和dlfi配置，优先级信息包不运载MLP报头和序号，并且MLP将分配在所有成员链接间的优先级信息包。结果，由CRTP压缩的数据包可能到达故障中在接收路由器。这禁止CRTP解压信息包报头并且强制CRTP丢弃数据包。

建议

推荐在套件的成员链接有同一个带宽。如果添加不同等的带宽链路到套件，将导致重拨抽签的数据包，将造成整体套件吞吐量减少。

推荐VIP2-50 (与8 MB SRAM)或更加高与这些分布式功能一起使用。

促销包编号和链路和内存要求

思科7500系列路由器的参考的分布式多链路点对点协议。

硬件与软件MLPPP或MLFR在7600个SIP线路卡

MLP和MLFR可以软件或基于硬件的。在硬件基于MLP或MLFR， Freedm提供多链路功能，并且MLP报头由FREEDM芯片添加。在软件基于MLP或MLFR， SIP线路卡CPU提供类似于VIP和FlexWan实施)的多链路功能(。

这些是运行硬件基于MLP或MLFR的限制和条件。

- 可以有仅最多每线卡336个套件和每安全状况评估(SPA) (Freedm) 168个套件。
- 可以有仅最多每个套件12 DS1/E1。
- 所有链路应该属于同样SPA (Freedm)。
- 在套件的所有链路应该运行以同一速度。
- TX分段大小可以是128， 256或者512。CLI配置的分段大小被映射对最近的支持的分段大小。

```
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分段大小可以是1到9.6 KB。
- 不可能支持思科所有权格式(MLFR)。

在硬件LFI，如果只有在套件的一条链路，并且，如果那是DS1/E1然后分段和交叉将由Freedm完成。

show ppp multilink输出显示硬件实现是否运行。

```
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的那么**show ppp multilink**输出有在输出中。

数据包的生活

Rx数据路径

1. 驱动程序接收的数据包。
2. 封装被检查：如下基本封装：在dmlp，入口接口的封装类型是ET_PPP。在dMLFR，入口接口的封装类型是ET_FRAME_RELAY。在dLFIoLL，入口接口的封装类型是ET_PPP。在

dLFIoFR，入口接口的封装类型是ET_FRAME_RELAY。在dLFIoATM，入口接口的封装类型是ET_ATM。在dDialer，封装类型是ET_PPP。另外封装处理：对于ET_PPP， NLPID发现。对于dmlp， NLPID是多链路。对于dLFIoLL，有要考虑的两件事：VoIP信息包—这些没有指示IP的一个MLP报头并且有NLPID。数据包— NLPID是多链路。对于dDialer，指示IP的数据包不会有一个MLP报头并且有NLPID。**注意：** 在这种情况下，您可以配置dCRTP (分布式 Compressed Real-Time Protocol)。如果那样，报头在进一步处理前被解压。

3. 对于ET_FRAME_RELAY，如果数据包接收的链路为dMLFR配置然后数据包为dMLFR处理
4. 对于dLFIoFR和dLFIoATM，封装类型分别为ET_FRAME_RELAY和ET_ATM，;但是在那内有PPP报头。PPP报头，如同dLFIoLL，指示数据包是否是语音数据包或数据包。如果dCRTP配置，报头在进一步处理前被解压。语音数据包立即交换。在交换前，分段的数据数据包将必须被重新组装。使用ET_PPP，您也许遇到PPP链路数据包;并且与ET_FRAME_RELAY，您也许遇到MLFR控制数据包。所有这些控制数据包被踢对处理的RP。
5. 根据上述解码，数据包被检查种类交换它要求。链路类型确定数据包是否是IP交换或MPLS交换的。数据包然后给对各自交换功能。
6. 使用捆绑与分布式功能一道， IP涡轮快速交换矢量窃取。因为数据包在成员链接，接收这执行;然而，必须对待这样在套件接收。您也需要检查被踢到主机的控制数据包。主要在dMLFR，有不是MLFR控制数据包的本地管理接口(LMI)数据包。对于这些，使用DLCI号码空间的一个不同的部分。每当DLCI在此空间解码下跌，数据包被踢至主机，因为被认可是LMI数据包。VoIP信息包(排队在低延时队列)交换，不用MLP报头的新增内容。当分段的数据数据包接收时，分布式功能能收到和重新组装数据包。重组进程在后面的章节解释。如果数据包一定标记交换的，然后通过对开关程序，在dmlp。否则，如果是IP交换，它通过对IP交换惯例。**注意：** 所有非IP信息包在dMLFR被踢主机。
7. **IP:**IP交换功能是普通到所有信息包。它做主要三件事：万一所有功能配置，执行必要处理数据包。并且，当使用时分布式拨号程序，请执行空闲计时器更新此处，当“Interesting Packets”接收时。参考dialer idle-timeout (interface)， dialer fast-idle (接口)和配置一拨号配置文件关于**空闲计时器**配置参数的详细信息。在75xx路由器上，邻接将指示出口接口的`tx_acc_ptr`。如果出口接口是虚拟访问接口， `tx_acc_ptr`是`NULL`。在这种情况下，请修复封装并且从`FIB hwidb`获得`tx_acc_ptr`。此查找和封装修复上升必要的在dLFIoFR和dLFIoATM。作为多链路捆绑一部分，在dLFIoLL，链路对待。**注意：** 数据包的TTL调节此处，并且IP分段的检查做。`mci_status`设置为所有信息包的`RXTYPE_DODIP`。
8. 当交换决定做，数据包准备从接口被发运。接口被检查确定是否支持本地交换。如果它，直接地通过fastsend被派出。否则，尝试做出到路由缓存交换机它。注意，万一QoS为接口配置，本地交换矢量由QoS窃取。在最终发送在接口外面前， HQF将排列数据包和进一步处理数据包。这是有dlfi的实际情形。对于dlfi，分段和交叉设置。QoS处理我们的分段惯例的调用并且插入分片数据包用在优先级队列将排队的语音数据包(如果LLQ配置)。这保证VoIP信息包不遭受要求的延迟通过链路发运巨大的数据包。

Tx数据路径

`vip_dtq_consumer`得到数据包并且获得接口号，获得`idb`。对应于`idb`fastsend惯例呼叫：

i) Fastsends

1. 在dmfr， fr_info结构从匹配if_index对fr_info的表获取。控制数据包被派出。帧头将给DLCI，将帮助您确定这是否是LMI数据包或数据包。帧头的DLCI字段用dmfr序号覆盖。独立的序号使用LMI和数据包。**注意：** 独立的序号使用分开的DLCI。
2. 在dmlp，控制数据包发送与优先级集对高。使用数据包，如果dCRTP配置，报头是被压缩的。包括定序的信息的VIP MLP报头被添加并且发送在成员链接外面。

3. 在dlfi， HQF截断通过接口将发送的数据包。如果它是语音数据包，语音数据包在优先级队列安置(如果LLQ配置)和发送在接口外面，不用MLP封装。使用数据包，它呼叫dlfi分段代码，返回片段对QoS代码，然后插入与优先级数据流，以便语音流量的延迟需求符合。并且，如果dCRTP配置，只有语音数据包的报头是被压缩的。当他们是，数据包报头被留下。
4. 在dDialer，在数据包被派出前，数据包分类为了重置输出链路的空闲计时器。这执行，在输出链路选择后，在几条链路一定对同一拨号程序情况下。报头没有被添加到拨号程序信息包。因此，定序和请重新召集数据包拨号接口不支持。

注意： 在dmlp、dDialer、dMLFR和dlfi与几条链路，流量转发的物理链路取决于链路的拥塞。如果链路拥塞，请移动向下条链路等等。(dMLFR、dmlp没有QoS和dDialer功能也选择根据字节数的链路放置在链路。它选择下条链路，如果当前链接已经传送字节其配额，在循环方式。此配额由链路的frag_bytes决定。对于拨号程序成员接口， frag_bytes设置为接口带宽的默认值。)

注意： 在出口VIP的接口的HQF配置中， HQF窃取dtq_consumer矢量。对出口VIP的数据包DMA'd首先通过HQF检查。如果QoS在出口接口配置， HQF起动处理数据包，在数据包是fastsent在接口外面前。

重组

无格式dDialer接口不支持重组并且定序。要启用此在拨号接口，在拨号接口的MLP将必须配置。如果这执行， Rx和Tx路径与dmlp路径是相同的。当数据包接收时，序号根据期望的序号核对。

- 如果序号配比：如果数据包是一未成碎片的数据包那么重组没有要求。继续进行更加进一步的交换步骤。如果数据包是片段，则请检查开始并且结束位并且修建数据包，当片段接收时候。
- 如果序号不配比：然后如果序号在序号内预计窗口放置它在排序的“未分配片段列表”。以后，当期望的序号没有接收时，此列表被检查，万一数据包存储此处。如果序号不在窗口内，请丢弃它并且报告“已接收丢失的片段”。如果超时发生以后，当等待时此数据包，接收方是重新同步的，并且用接收的下一个信息包再开始。

总计那些案件，正确地被定购的信息包流被发送在此接口外面。如果片段接收，一完整数据包形成然后被派出。

配置，正在验证和调试分布式功能

此部分包括是可用验证和调试其中每一个分布式功能的**显示**和调试指令。

配置和正在验证的dmfr

MFR配置示例

```
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.
```

注意： MFR接口是类似另一个FR接口并且支持大多数FR配置。

```
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.
```

验证MFR在RP的套件状态

```
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
```

这表明两个接口正确地被添加，并且一个接口未协商MLFR LIP消息。

要获得关于MFR套件和成员链接的更多信息，请发出此命令：

```
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调试指令

这些调试是有用的排除故障链路不添加到套件的问题。

debug frame-relay multilink control

注意： 当特定MFR接口或Serial interfaces没有指定时，所有MFR的此enable (event)调试连接。如果路由器有很大数量的MFR链路，这可以压倒多数。

要调试接收在RP的MFR数据包，以及调试MFR控制活动，此调试是有用的：

debug frame-relay multilink

注意： 在大流量下，这能淹没CPU。

验证dMLFR在LC的套件状态

show frame-relay multilink

注意： 目前，这不是可用的在LC，但是很快将被添加。到那时，使用**show ppp multilink**。

debug frame-relay multilink

配置和正在验证的dMLP/dLFIoLL

多链路PPP配置

debug frame-relay multilink

在Serial interfaces下的配置示例：

```
debug frame-relay multilink
```

注意： m1命令的ppp chap hostname不确实意味着CHAP认证启用。如果那里是超过在同一两路由器之间的一个多链路捆绑在此命令的字符串M1作为end-point-discriminator和只要求。在这种情况下，属于套件的所有链路不应该有同一个端点分辨器和属于一个不同的套件的两条链路应该有同一个端点分辨器。

可选配置参数

[no] ppp multilink interleave

这启用在多链路捆绑的交叉。这与模块化QoS CLI一道工作。而其他数据包将分段并且传送与MLP顺序和报头，高优先级数据包将传送，不用MLP顺序和报头的新增内容。

注意： 当插入启用与超过一条链路时，很可能，高优先级数据流将获得重新命令。当插入是启用或禁用的时，套件的重置在线卡要求获得它激活。

```
ppp multilink mrru local value
```

这指定在多链路的最大接收单元;至此大小的数据包将由多链路接口接受。此处大小排除MLP报头。

```
ppp multilink mrru remote value
```

这指定远程终端应该支持的最低的MRRU。如果远程终端MRRU比此值是较少，则套件协商将发生故障。

```
ppp multilink fragment delay seconds
```

这以数据分段(毫秒)指定允许延迟造成的毫秒。换句话说，值延迟用于计算允许的最大分段大小。分布式实施与Cisco IOS实施有所不同用这些方式：

1. 除非插入启用，分段没有进行。
2. 使用变化的带宽链路，选择的分段大小根据最少带宽接口。

```
ppp multilink fragment disable
```

此命令不添加在分布式实施的任何功能。当插入启用时，分段只发生;并且，当插入启用时， ppp multilink fragment disable命令忽略。

验证dmlp在RP的套件状态

```
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. 当套件在分布式模式时，这在**show ppp multilink**中输出显示：由于某种原因否则，然后没有分配套件。
2. 当**ppp multilink interleave**配置并且启用在线卡时， **show ppp multilink**输出包括`dlfi`where: —指示传送并且接收的计数片段。—指示传送或接收，无需被分段的计数数据包。—指示被重新组装完整的信息包的数量。当插入没有启用时，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
```

在前一个示例的分段大小是760个字节。

验证dmlp在LC的套件状态

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，序号维护根据每DLCI基础，用在用于LMI DLCI的套件的序号。

字段	说明
dmlp_ipc_config_count	IPC消息编号多链路或MLFR配置的LC接收的
dmlp_bundle_count	MLP和MLFR编号在LC捆绑
dmlp_ipc_fault_count	在LC导致失败配置消息的编号。应该是0;如果它是非零那么也许有问题。
	指示idb对tag_optimum_fs和idb对用于标记交换的ip2tag_optimum_fs向量。
board_encap	指示是使用的添加2字节的板封装的board_encap矢量，如果有信道化的链路在7500平台。如果链路包含非信道化的接口，应该是NULL。
max_particles	在重组缓冲可以保持微粒的最大
mrru	接受，无需考虑MLP封装的最大大小数据包。不可适用为MLFR接口。
seq_window_size	序号的最大窗口尺寸
working_pak	指示当前朴在重组下。NULL，如果。
working_pak_cache	对使用重组的静态朴的指示器。这，当第一不完整数据包由套件时，接收分配。
una_frag_list	在重组队列的首先进入。如果条

	目不是NULL并且不更改，表明计时器不运行软件问题。
una_frag_end	在重组队列的最后一项
rcved_end_bit	表明套件接收末端位，因此它寻找开始位。
is_lost_frag	如果片段被宣称丢失，是真的。当与期望的顺序的一个片段接收时，这清除。
resync_count	指示次数接收方用发射器是不同步的并且通过开始必须再同时与为时接收的程序化的片段。
	表明重组超时出现和数据包从重组队列处理。
timer_start	次数重组计时器开始
timer_running	指示重组计时器是否运行。
timer_count	指示重组计时器超时的次数。
next_xmit_link	下一个信息包将传送的链路
	指示当前的成员的位域。
	用于所有分组没的字段。指示哪些成员链接没有拥塞。
dmlp_orig_pak_to_host	矢量曾经踢数据包到RP。
dmlp_orig_fastsend	在MLP或MLFR前的原始驱动程序fastsend修改了驱动程序的fastsend。
	丢失片段的编号(接收方没有收到这些片段)。这，当更新被发送到主机时，周期地清除。
	接收出于预计故障片段的编号。这，当更新被发送到主机时，周期地清除。
	丢弃的片段编号，因为一完整数据包不可能做
	认为丢失片段的编号接收。这表明交互相联延迟比30毫秒dmlp重组超时极大。

配置的和正在验证的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dLFIoFR/ATM在RP的套件状态

```
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： 只有当ppp multilink interleave配置在虚拟模板下，套件将变得分配;没有此命令，不会分配套件。

验证dLFIoFR/ATM在LC的套件状态

要验证dlfi正确地的确工作在LC，发出此命令：

```
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。分段将完成在WFQ分支层blt。

配置和正在验证的dDDR

当您启用在拨号接口时，和ip route-cache的ip cef distributed分配的全局配置分布式DDR激活。

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
```

没有分布式DDR的其他特别配置。进一步配置跟随普通的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
```

告诉我们使用的拨号程序种类。ISDN暗示传统拨号程序配置，并且暗示Dialer Profile配置。指示拨号程序的现状。一无关联的拨号接口的状态。重置，每当关注数据流被看到。如果此计时器超时，接口将立即断开。是可配置参数。欲知详情，参考配置与拨号配置文件的点对点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
```

显示的变量是相同的象那些为dmlp。

调试dmlp和dDDR

调试可用在RP

dDDR

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

发出此命令调试控制路径功能类似呼叫建立等等。欲知详情，参考debug dialer events。

```
debug ip cef dialer
```

发出此命令调试CEF相关的拨号事件。欲知详情，参考Dialer CEF。

调试可用在LC

dmlp

控制路径调试：**调试多链路事件**

数据路径调试：**调试多链路片段**

数据路径和控制路径错误调试：**调试多链路错误**

调试在SIP线路卡的dmlp

转存根据CI的数据包：数据包和控制数据包在根据控制ci和顺序ci的线路卡可以被转存。

hw-module测验*subslot_num*转储ci *CI-NUM* [rx|tx] *num_packets_to_dump*

同边可以如此获取：

```
!--- 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，您必须获取vc，可以从**show interface serial *CT3_interface_name***输出获取。

现在CI信息可以从SPA控制台得到。首先请重定向SPA控制台命令输出对RP用**spa_redirect RP ct3_freedm336**命令。

spa_ct3_test freedm显示linkrec vc命令显示必要的CI信息。

dmfr

控制路径调试：**调试dmfr事件**

数据路径调试：**调试dmfr数据包**

数据路径和控制路径错误调试：**调试dmfr错误**

转存根据CI的数据包：请参阅dmlp。

dlfi

控制路径调试：**调试dlfi事件**

数据路径调试：**调试dlfi片段**

数据路径和控制路径错误调试：**调试dlfi错误**

dDDR

没有特殊调试命令;您应该使用dmlp调试。

在dLFIoLL的情况下， dmlp和dlfi调试也许必须使用。这些调试没有条件的，并且，因此，为所有套件触发。

常见问题

1. **什么是dmlp ？**dmlp为Distributed多链路PPP是短的(如RFC1990所述)。此功能乘分布式平台支持，类似Cisco 7500系列和7600系列。dmlp允许您结合T1/E1线路—在Cisco 7500系列路由器的VIP或在7600系列路由器的FlexWan —到有复合带宽多条T1/E1线路的套件。这允许客户增加在T1/E1之外的带宽，不用需要采购T3/E3线路。
2. **什么是“分布式”在dmlp ？**期限“分配了”暗示数据包交换由VIP而不是RSP完成。为什么？RSP交换功能相当被限制，并且它有许多重要工作执行。的VIP有能力在交换信息包上卸载从RSP的此活动。基于RSP的Cisco IOS仍然管理链路。捆绑创建和卸载由RSP完成。另外，

PPP控制层面处理由RSP仍然完成，包括处理所有PPP控制数据包(LCP、验证和NCP)。然而，一旦套件设立， MLP数据包处理被移交对交换的VIP由内置CPU。dmlp引擎(在VIP)处理所有MLP步骤，包括分段、交叉、封装，负载均衡在多条链路中和入站片段排序和重组。在7500系统的VIP完成的功能由FlexWan/Enhanced-FlexWAN完成在7600基于系统。

3. **如何知道是否分配套件？**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. **如果我升级对RSP16或SUP720， dmlp性能是否将是更加好？**不能。dmlp (或任何分布式功能)交换性能依靠有问题的VIP或的FlexWan。例如， VIP6-80的性能比与VIP2-50的性能好。
5. **能以此功能使用哪些PA ？**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仅)
6. **多少条链路可以配置在一个套件？**有许多面对此答案。主要的瓶颈是线卡(VIP/FlexWAN/Enhanced-FlexWAN2)的CPU电源。硬限制是每个套件56条链路，但是您不能配置那些许多的许多次(和有流量交换)，由于CPU电源或有限缓冲区。这些编号根据此指南(根据CPU和内存在VIP/FlexWAN/Enahnced-FlexWAN2) ：VIP2-50 (与4MB SRAM)最大T1 = 12VIP2-50 (与8MB SRAM)最大T1 = 16VIP4-80最大T1 = 40VIP6-80最大T1 = 40FlexWan最大T1 =短期将更新Enhanced-FlexWAN最大E1s =每个海湾(聚合42 E1s 21 E1s每线卡)
7. **有没有在性能上的一个变化，如果我配置与3 T1的3个套件中的每一个或与9 T1的1个套件？**没有在性能上的变化，如证明在实验室测试。然而，与在一个套件的很大数量的T1 (24或28 T1请说在一个套件的)，有关于用尽缓冲区的问题。其高度推荐您没有超过8个成员链接(T1/E1)在一个套件。
8. **如何确定套件的带宽？**套件的带宽不将配置。其聚合带宽所有成员链接。如果有4 T1在套件，则套件的带宽是6.144Mbps。
9. **更加好是哪些？CEF负载平衡或dmlp ？**没有简单的回答对此。您的需要决定哪个是更加好。**MLP专业人员：**CEF负载均衡是仅可适用的对IP数据流。MLP平衡所有流量发送套件。MLP维护数据包排序。IP是宽容重拨，因此这可能不要紧对您;实际上，在维护涉及的附加费用定序可能是原因避免MLP。IP供可能传送故障中的数据包的网络使用，并且任何使用IP应该能处理重拨。然而，尽管此事实，实际情况是重拨能仍然提出真正的问题。MLP提供对对等系统的单个逻辑连接。多链路捆绑支持QoS。MLP提供动态带宽功能，用户能添加或删除根据当前需要的成员链接。MLP能捆绑链路更大的量，而CEF负载均衡对6个并行IP路径被限制。单个流的CEF负载均衡对一个T1限制最大带宽所有特定传。例如，使用语音网关的客户只能安排与同一源和目的很多呼叫，并且，因此，使用一个路径。**MLP缺点：**MLP添加额外的开销到每数据包或帧MLP强化中央处理;dmlp是强化中央处理的线卡。
10. **如何能配置在两路由器之间的多个套件？**多链路确定哪个套件链路将加入基于对等体的名称和端点分辨器。要创建在两个系统之间的多个明显的套件，标准方法将强制某些链路不同地识别。推荐的方法是使用**ppp chap hostname name命令**。
11. **能否有从不同的PA的成员链接？**不能。如果要运行dmlp，则不支持。然而，如果成员链接从不同的PA被添加，然后控制不再给对RSP和没有其dmlp。MLP仍然作用，但是dmlp的好处去。
12. **能否混合从两个海湾的成员链接？**不能。如果要运行dmlp，则不支持。然而，如果成员链接从不同的PA被添加，然后控制给对RSP，并且它不再是dmlp。MLP仍然作用，但是dmlp的

好处去。

13. **能否有在不同的VIP或FlexWANs间的成员链接？**不能。如果要运行dmlp，则不支持。然而，如果成员链接从不同的PA被添加，然后控制不再给对RSP和没有其dmlp。MLP仍然作用，但是dmlp的好处去。
14. **能否有在不同的端口间的成员链接从单个PA ？**(例如，从PA-MC-2T3+的每个CT3端口的一个成员链接。)可以。只要它是从同样PA，没有问题。
15. **能否捆绑T3或E3端口？**不能。仅DS0、n*DS0， T1和E1速度允许与dmlp为7500/VIP、7600/FlexWAN和7600/FlexWAN2。**注意：** 分布式MLPPP为成员链接仅支持配置以T1/E1或subrate T1/E1速度。信道化的STM-1/T3/T1接口也支持dMLPPP以T1/E1或subrate T1/E1速度。分布式MLPPP不为成员链接支持配置以clear-channel T3/E3或更高的接口速度。
16. **什么是"重拨的"片段？**如果已接收片段或数据包不匹配期望的序号，则计数器被增加。对于变化的数据包大小，这一定发生。对于固定尺寸的数据包，这能也发生，因为PA驱动程序处理在一条链路接收，并且不努力去做在循环方式的数据包(和在dmlp执行，当传送数据包)时。reordered不含义包丢失。
17. **什么是"丢失的"片段？**每当片段或数据包接收的有故障，并且您发现故障中片段或数据包在所有接收链路，计数器被增加。另一个案件是，当故障中片段在列表时存储，并且达到限制(决定根据在VIP的SRAM，并且什么为套件分配)，计数器被增加的丢失的片段，并且在列表的下个序号为处理被采取。
18. **dmlp如何检测丢失的片段？**序号：如果等待一个片段用序号N到达，并且所有链路高于N收到与序号的一个片段，您知道必须丢失片段N，因为没有办法可能在同一条链路的更高的被编号的片段后合法到达。超时：如果坐太长等待片段，您最终将宣称它如丢失并且移动。重组缓冲溢出：如果等待片段N到达和同时其他片段(用序号高于N)在某些到达链路，则您必须停放在重组缓冲的那些片段，直到片段N出现。有限制对多少您能缓冲。如果缓冲区溢出，您再宣称片段N如丢失，并且恢复处理与什么在缓冲区。
19. **什么"丢失的已接收？"**有丢失的已接收片段或数据包的两个可能的来源：如果已接收片段或数据包是在期望的顺序范围窗口外面，数据包通过标记它丢弃如丢失的已接收。如果已接收片段或数据包在期望的顺序范围窗口内，但是不能分配信息包报头到re-parent此数据包，则数据包丢弃并且被标记作为丢失的已接收。
20. **加密支持与dmlp ？**不能。
21. **是否支持PFC报头压缩？**不，不在分布式路径。没有推荐远端的路由器配置PFC报头压缩，因为我们跌倒回到非分布式的模式，如果我们收到被压缩报头帧或数据包。如果要继续运行dmlp，必须禁用PFC报头压缩在两端。
22. **软件压缩支持与dmlp ？**不，因为软件压缩不会在分布式路径工作。
23. **传输端支持分段？**不与香草dmlp。没有关于接收片段的问题与香草dmlp，但是在传输端，分段不发生。传输端分段，当**ppp multilink interleave**在dmlp接口时，配置支持。
24. **能否ping MLP套件的成员链接？**不，您不能配置在成员链接的一个IP地址。
25. **有没有在链路MTU和MLP分段大小的任何从属关系？**不能。MTU大小与MLP分段大小无关，除一个MLP片段，类似其他帧，不可以超出串行链路的MTU大小的明显的限制之外。
26. **配置在一个对的两个MLP套件路由器之间是否是可能的？**是，这是可能的。然而，这能导致被削弱的负载均衡。可以是有用的在试验床，模拟超过使用两路由器的两路由器，但是它没有任何明显的真实世界的值。去一普通的对等体的所有链路在同一个套件*必须*放置。根据定义，套件是去特定对等体的套链路。"对等体"由在LCP和认证阶段期间，它提供的用户名和端点分辨器值识别。如果尝试创建在两路由器之间的多个套件，则含义您比单个对等体尝试做每个路由器化妆象是更多对其副本。他们必须识别适当地。
27. **成员链接能否有不同的排队算法？**与套件需要涉及的所有排队机制应用在套件级别和不在成员链路级。然而，配置队列算法不应该影响数据包如何交换在套件外面。
28. **当dmlp在Cisco 7500时，启用tx-quque-limit为什么设置到26作为成员链接的默认多链路捆绑的？**对于带宽T1/E1所有Serial interfaces， tx-queue-limit是大约4或5。当您在多链路时一起

捆绑T1s/E1s，带宽为套件将增加。由于交换将发生基于带宽MLP接口，您需要增加tx-queue-limit成员链接。仅一成员链接，呼叫主链路，使用交换，因此，其tx-queue-limit需要增加。并且，此值是一经验主义一个选定的在测试然后调整以后对此值。一般来说，部署没有超过4对6T1/E1链路在套件。值为26能完全顾及6条到8条T1/E1链路，并且此值选择。

29. **什么是差分延迟和其值在dmlp实施？**dmlp支持30毫秒差分延迟。那含义片段是否每次是接收的T，并且有故障(期待序号100，但是我们接收101)。如果序号100没有接收直到T+30毫秒，100将被宣称丢失，并且，如果能开始处理从101，您会执行那。万一不能从101开始(如果它是一个中间片段)，您会寻找有开始从那里的片段的片段(例如， 104)和开始。
30. **当数据包被分段在与多链路的IP级别在7500时，什么发生？**如果数据包被分段在IP级别，则他们传输，不用在半成品跳的重组，但是被重新召集在目标路由器。
31. **当数据包被分段在7500时的MLP级别什么发生？**如果数据包被分段在MLP级别，并且，如果重新组装的信息包比MRRU极大，然后数据包在多链路丢弃。transmit-side仅dmlp支持分段与dlfi。只有当packet_size比frag_size是极大和较少比MRRU，数据包被分段在MLP级别。如果他们没有被分段在MLP级别的数据包更多比MRRU被发送，并且，如果没有被分段在IP级别，然后另一端下降所有信息包大小，因为数据包比MRRU是更多。
32. **MRRU如何计算？**MRRU根据这些首选计算：对于进来新成员的链路， MRRU再协商在LCP级根据在成员链接配置的MRRU。在与**interface命令PPP多链路的mrru的**链路接口配置的值。如果没配置，值从**PPP多链路mrru** on命令的配置继承了parent接口。如果两个值存在，链路接口值有优先。默认MRRU 1524。

调试增强

这些增强将占去今后。规划不是完成。

- Enable (event) **debug frame-relay多链路**on命令LC。
- 提高当前调试CLIs每数据包接口和指定的编号。
- 对于dDDR，不支持QoS功能。这可以用适当的商业案例仅占去。

相关信息

- Dialer CEF
- 使用 Dialer Profile 配置对等 DDR
- MPLS —多链路PPP支持
- 思科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)
- 在租用的线路的Distributed Link Fragmentation and Interleaving
- 带有服务质量控制（LLQ/IP RTP 优先级、LFI、cRTP）的 VoIP-over-PPP
- 排除故障TechNotes - Cisco 7500系列路由器
- 路由器产品支持页- Cisco系统
- 技术支持和文档 - Cisco Systems