多机箱多链路PPP (MMP) (第2部分)

目录

简介

本文继续描述多链路PPP的支持在“堆叠”或多机箱环境(有时呼叫MMP，*多机箱多链路PPP的*)，在Cisco系统的接入服务器平台。

本文是一个两部分文档的一部分两。本文的参考的第一部分欲知更多信息。

先决条件

前提对于本文在本文的第一部分中给。

示例

堆栈（带拨号程序）中的 AS5200

当拨号程序在物理接口时配置，没有需要指定虚拟模板接口。虚拟访问接口作为无源接口，支持在拨号接口和物理接口之间关联与拨号接口。

简而言之，您只需要定义堆栈组名称、普通的密码和堆栈组成员在所有堆栈成员间。如以下示例所显示，虚拟模板接口没有定义，：

```
systema#config
  sgbp group stackq
  sgbp member systemb 1.1.1.2
  sgbp member systemc 1.1.1.3

  username stackq password therock

  int dialer 1
  ip unnum e0
  dialer map .....
  encap ppp
  ppp authen chap
  dialer-group 1
  ppp multilink

  controller T1 0
  framing esf
  linecode b8zs
  pri-group timeslots 1-24

  interface Serial0:23
  no ip address
  encapsulation ppp
  dialer in-band
  dialer rotary 1
  dialer-group 1
```

以下示例是从E1控制器：

```
systema#config
  sgbp group stackq
  sgbp member systemb 1.1.1.2
  sgbp member systemc 1.1.1.3

  username stackq password therock

  int dialer 1
  ip unnum e0
  dialer map .....
  encap ppp
  ppp authen chap
  dialer-group 1
  ppp multilink

  controller T1 0
  framing esf
  linecode b8zs
  pri-group timeslots 1-24

  interface Serial0:23
  no ip address
  encapsulation ppp
  dialer in-band
  dialer rotary 1
  dialer-group 1
```

在捆绑接口创建后，用从拨号接口的仅PPP命令克隆。随后的设计的PPP链路也被克隆用从拨号接口的PPP命令。图3显示拨号接口如何坐在捆绑接口顶部。比较它图2，没有拨号接口。

PRI和BRI默认情况下是拨号接口;PRI配置没有一明确拨号程序(通过**dialer rotary命令**)仍然是在Serial0:23的一拨号接口，如显示由以下示例：

```
systema#config
  sgbp group stackq
  sgbp member systemb 1.1.1.2
  sgbp member systemc 1.1.1.3

  username stackq password therock

  int dialer 1
  ip unnum e0
  dialer map .....
  encap ppp
  ppp authen chap
  dialer-group 1
  ppp multilink

  controller T1 0
  framing esf
  linecode b8zs
  pri-group timeslots 1-24

  interface Serial0:23
  no ip address
  encapsulation ppp
  dialer in-band
  dialer rotary 1
  dialer-group 1
```

图 3：堆栈组– stackq –包括systema、systemb和systemc。systema链路在拨号接口配置。

使用卸载服务器

systema被选定卸载服务器(使用**sgbp seed-bid命令**)。其他堆栈组成员必须用**sgbp seed-bid default命令**定义(或，如果不定义了thesgbp**种子出价**命令，默认为此)。

```
systema#config
  multilink virtual-template 1
  sgbp group stackq
  sgbp member systemb 1.1.1.2
  sgbp member systemc 1.1.1.3
  sgbp seed-bid offload
  username stackq password therock

  interface virtual-template 1
  ip unnumbered e0
  :

  ppp authen chap
  ppp multilink
```

图 4：systema作为卸载服务器。

卸载带物理接口的服务器

如果指定卸载服务器也有希望的物理接口(例如， PRI)服务telco搜索组和其他堆栈成员一样，您能配置它通过结合在题为在斯塔克的AS5200的本文的部分显示的配置(与拨号程序)和使用卸载服务器如此执行。

被卸载的设计的PPP链路和其捆绑接口依靠配置源的虚拟模板。有*第一条链路的*连接到达在View (物理设备连接对拨号接口和捆绑接口和所有随后的设计的PPP链路的配置源是拨号接口配置。因此，这些变化共存，从属在第一条链路到达的堆栈成员。

此配置不推荐的归结于在拨号程序和虚拟模板接口要求的配置复杂性。

异步、串行和其他非拨号接口

当您能配置异步，并且时串行设备作为拨号接口(在恢复对在斯塔克的AS5200 (与拨号程序)情况下，如本文的该部分所显示)，您可以选择支持多机箱MP，不用异步，序列的任何拨号程序配置和其他非拨号接口。所有配置来源在虚拟模板接口然后定义，如下所示。

```
#config
  multilink virtual-template 1
  sgbp group stackq
  sgbp member systemb 1.1.1.2
  sgbp member systemc 1.1.1.3
  username stackq password therock
  interface virtual-template 1
  ip unnumbered e0
  :
  ppp authen chap
  ppp multilink

  int async 1
  encap ppp
  ppp multilink
  ppp authen chap
  :

  line 1
  login
  autoselect ppp
  autoselect during-login
  speed 38400
  flow hardware
```

从多底盘拨出

目前，因为第二层转发协议不支持拨出，多机箱配置不**支持拨出**。

结果，没有办法启动的卸载服务器(其中路由在拨号配置文件被伪装，等等)在前端堆栈成员的一个拨号在同一个堆栈组中。在前端堆栈成员必须安装所有被伪装的路由，因为这些是那个物理拨号连接(例如PRI)。

一些应急方案如下：

- 当**sgbp ppp-forward命令**在前端堆栈成员时发出，这意味着所有PPP和PPP多链路呼叫自动地转发给堆栈组竞标协议(SGBP)投标赢利地区，例如卸载服务器。您在网络接入服务器(NAS)必须取决于拨号，并且请让IP路由收敛(仅IP)采取其课程。例如，拨号1.1.1.1，请放置此地址在拨号映射语句在NAS并且放置静态路由在NAS，如下：

  ```
  #config
    multilink virtual-template 1
    sgbp group stackq
    sgbp member systemb 1.1.1.2
    sgbp member systemc 1.1.1.3
    username stackq password therock
    interface virtual-template 1
    ip unnumbered e0
    :
    ppp authen chap
  ```

```
ppp multilink

int async 1
encap ppp
ppp multilink
ppp authen chap
:

line 1
login
autoselect ppp
autoselect during-login
speed 38400
flow hardware
```

当拨号连接给远端对等体时， PPP连接形成在远端对等体和卸载服务器之间。前端堆栈成员完全被绕过。在卸载服务器的PPP然后安装主机路由给对等体— 1.1.1.1。这时，因为路由度量趋向路由那里， IP路由协议从到主机路由聚合在卸载服务器。**注意：** 路由聚合导致延迟。

- 当**sgbp ppp-forward命令**在前端堆栈成员时没有定义，这意味着仅PPP多链路呼叫自动地转发给SGBP投标赢利地区，例如卸载服务器。因此，从前端堆栈成员的一拨号程序远端对等体的跨过外端点和远端对等体之间的PPP连接—同一种行为，好象NAS没有作为堆栈组的部分。**注意：** 这发生，只要连接是直通而不是PPP (PPP多链路)。

向多底盘拨号

如果有IP路由(例如增强的内部网关路由选择协议(EIGRP)和开放最短路径优先(OSPF))流在客户端和最终赢取投标的堆栈成员之间(例如卸载服务器)，这跟随的一些个提示：

防止安装在客户端的已连接路由

配置1.1.1.2是NAS的客户端1.1.1.2 (透明帧转发器)的地址，如下所示。

```
#config
  multilink virtual-template 1
  sgbp group stackq
  sgbp member systemb 1.1.1.2
  sgbp member systemc 1.1.1.3
  username stackq password therock
  interface virtual-template 1
  ip unnumbered e0
  :
  ppp authen chap
  ppp multilink

  int async 1
  encap ppp
  ppp multilink
  ppp authen chap
  :

  line 1
  login
  autoselect ppp
  autoselect during-login
  speed 38400
  flow hardware
```

如果有EIGRP，例如，运作在客户端和卸载服务器之间，在卸载服务器的路由表指示那达到

1.1.1.2路由应该通过虚拟访问接口。这是因为点到点协议IP控制协议(IPCP)在客户端安装已连接路由1.1.1.2对BRI接口。EIGRP然后通告此路由到在PPP会话的卸载服务器(在L2F)。在卸载服务器的EIGRP因而指示那达到1.1.1.2，它应该路由对客户端—客户端的路由1.1.1.1是对虚拟访问接口。

现在，您有为客户端注定的一数据包1.1.1.1。IP路由发送数据包对虚拟访问接口。虚拟访问接口封装IP/User数据协议(UDP)/L2F/PPP封装和发送数据包对L2F NAS — 1.1.1.2。一切这时是正常。然后，而不是发送数据包通过(例如)以太网接口， IP路由通过再虚拟访问接口发送它。这是因为路由表指示那达到NAS，它必须通过客户端。这创建路由环路和有效禁用在L2F通道的输入和输出。

要防止此，请勿允许IPCP安装在客户端的一已连接路由。

注意： 只有当您有运作在客户端和思科家用网关之间时的某个IP路由协议这适合于。

客户端配置如下：

```
#config
  multilink virtual-template 1
  sgbp group stackq
  sgbp member systemb 1.1.1.2
  sgbp member systemc 1.1.1.3
  username stackq password therock
  interface virtual-template 1
  ip unnumbered e0
  :
  ppp authen chap
  ppp multilink

  int async 1
  encap ppp
  ppp multilink
  ppp authen chap
  :

  line 1
  login
  autoselect ppp
  autoselect during-login
  speed 38400
  flow hardware
```

在客户端的拨号图

当客户端拨号对多机箱环境时，总是请定义拨号程序给多链路捆绑的每个潜在的赢利地区。例如，如果有在多机箱堆叠的四个卸载服务器，那里应该在客户端定义的四拨号图。

例如：

```
#config
  multilink virtual-template 1
  sgbp group stackq
  sgbp member systemb 1.1.1.2
  sgbp member systemc 1.1.1.3
  username stackq password therock
  interface virtual-template 1
  ip unnumbered e0
  :
  ppp authen chap
```

```
  ppp multilink

  int async 1
  encap ppp
  ppp multilink
  ppp authen chap
  :

  line 1
  login
  autoselect ppp
  autoselect during-login
  speed 38400
  flow hardware
```

在本例中， 1.1.1.3是一个卸载服务器。

因为有dialer map匹配，为1.1.1.2路由注定的数据包对BRI和拨号程序拨号目的地。卸载服务器1.1.1.4实际上赢取投标，并且PPP会话设想那里。EIGRP交换在客户端和卸载服务器之间。在客户端的IP路由表充满对BRI0的一个路由1.1.1.4 (卸载服务器)。现在，在客户端，为1.1.1.4注定的数据包路由对BRI0。尽管没有拨号程序匹配，拨号器，然而，不能拨号。

注意： 总是请定义所有潜在的SGBP投标赢利地区的拨号图客户端的，每当访问卸载服务器是客户端的需求。

配置和限制

- 企业j-image为多机箱MP要求。
- 仅一个堆栈组可以为每接入服务器定义。
- 高延时广域网连接堆栈成员之间，导致MP重组延迟，可能造成多机箱MP是效率低的。
- 接口为PRI、[M] BRI、序列和异步装置支持。
- 不支持拨出。

配置每协议接口配置

实际上，请勿配置在虚拟模板的特定协议地址。

```
#config
  multilink virtual-template 1
  sgbp group stackq
  sgbp member systemb 1.1.1.2
  sgbp member systemc 1.1.1.3
  username stackq password therock
  interface virtual-template 1
  ip unnumbered e0
  :
  ppp authen chap
  ppp multilink

  int async 1
  encap ppp
  ppp multilink
  ppp authen chap
  :

  line 1
```

```
  login
  autoselect ppp
  autoselect during-login
  speed 38400
  flow hardware
```

虚拟模板接口担当任何数量的虚拟访问接口动态地被克隆的模板。您不应该指定对虚拟模板接口的一个单个接口的协议特殊化地址。由于IP地址一定是唯一为每个网络接口，指定在虚拟模板接口的一个唯一IP地址是不正确的。反而，请执行以下：

```
#config
  multilink virtual-template 1
  sgbp group stackq
  sgbp member systemb 1.1.1.2
  sgbp member systemc 1.1.1.3
  username stackq password therock
  interface virtual-template 1
  ip unnumbered e0
  :
  ppp authen chap
  ppp multilink

  int async 1
  encap ppp
  ppp multilink
  ppp authen chap
  :

  line 1
  login
  autoselect ppp
  autoselect during-login
  speed 38400
  flow hardware
```

配置全局协议配置

拨号到单个接入路由器并且当前盼望接入服务器有一个唯一全局地址的客户端(例如DECNet)实际上拨号给包括几接入服务器的多机箱多链路堆栈组。在这种情况，请终止确定性堆栈组在单个接入服务器。执行那，发出**sgbp seed-bid offload命令**在指定接入服务器(或指定最高的投标)。

排除故障

要执行的第一件事，如果有问题是回到单个堆栈成员，禁用其他堆栈成员。然后请测试您的PPP多链路连接并且通过通常质询握手验证协议(CHAP)验证和接口配置错误的在配置方面等等。当您是满足的时它工作，启用其他堆栈成员，然后如下进行：

1. 确保SGBP是正在运行的。
2. Debug ppp多链路。
3. 调试VPN和L2F。

确保 SGBP 已启动并正常运行

发出**show sgbp命令**确保，所有成员国是活跃的。否则，请寻找IDLE、AUTHOK或者活动状态。如被提及以前， IDLE是有意不激活的所有远程栈成员的一个有效状态。

如果如上所述查找一问题，请打开**debug sgbp hellos**和**debug sgbp error命令**。在两个堆栈成员之间的验证，例如在systema和systemb，应该如下(在systema) ：

```
systema# debug sgdp hellos

%SGBP-7-CHALLENGE: Send Hello Challenge to systemb group stackq
%SGBP-7-CHALLENGED: Hello Challenge message from member systemb (1.1.1.2)
%SGBP-7-RESPONSE: Send Hello Response to systemb group stackq
%SGBP-7-CHALLENGE: Send Hello Challenge to systemb group stackq
%SGBP-7-RESPONDED: Hello Response message from member systemb (1.1.1.2)
%SGBP-7-AUTHOK: Send Hello Authentication OK to member systemb (1.1.1.2)
%SGBP-7-INFO: Addr = 1.1.1.2 Reference = 0xC347DF7
%SGBP-5-ARRIVING: New peer event for member systemb
```

systema发送一CHAP风格挑战并且收到从systemb答复。同样地， systemb派出挑战并且收到从systema答复。

如果验证发生故障，您看到：

```
systema# debug sgdp hellos

%SGBP-7-CHALLENGE: Send Hello Challenge to systemb group stackq
%SGBP-7-CHALLENGED: Hello Challenge message from member systemb (1.1.1.2)
%SGBP-7-RESPONSE: Send Hello Response to systemb group stackq
%SGBP-7-CHALLENGE: Send Hello Challenge to systemb group stackq
%SGBP-7-RESPONDED: Hello Response message from member systemb (1.1.1.2)
%SGBP-7-AUTHOK: Send Hello Authentication OK to member systemb (1.1.1.2)
%SGBP-7-INFO: Addr = 1.1.1.2 Reference = 0xC347DF7
%SGBP-5-ARRIVING: New peer event for member systemb
```

这意味着stackq的远程systemb密码不匹配在systema定义的密码。

```
systema# debug sgdp hellos

%SGBP-7-CHALLENGE: Send Hello Challenge to systemb group stackq
%SGBP-7-CHALLENGED: Hello Challenge message from member systemb (1.1.1.2)
%SGBP-7-RESPONSE: Send Hello Response to systemb group stackq
%SGBP-7-CHALLENGE: Send Hello Challenge to systemb group stackq
%SGBP-7-RESPONDED: Hello Response message from member systemb (1.1.1.2)
%SGBP-7-AUTHOK: Send Hello Authentication OK to member systemb (1.1.1.2)
%SGBP-7-INFO: Addr = 1.1.1.2 Reference = 0xC347DF7
%SGBP-5-ARRIVING: New peer event for member systemb
```

这意味着systema有通过TACACS+或密码定义的本地或一个用户名。

一般来说，请定义在所有堆栈成员间的一普通的机密堆栈组stackq。您能定义他们本地或通过TACACS+。

在每个堆栈成员定义的本地用户名是：

```
systema# debug sgdp hellos

%SGBP-7-CHALLENGE: Send Hello Challenge to systemb group stackq
%SGBP-7-CHALLENGED: Hello Challenge message from member systemb (1.1.1.2)
```

```
%SGBP-7-RESPONSE: Send Hello Response to systemb group stackq
%SGBP-7-CHALLENGE: Send Hello Challenge to systemb group stackq
%SGBP-7-RESPONDED: Hello Response message from member systemb (1.1.1.2)
%SGBP-7-AUTHOK: Send Hello Authentication OK to member systemb (1.1.1.2)
%SGBP-7-INFO: Addr = 1.1.1.2 Reference = 0xC347DF7
%SGBP-5-ARRIVING: New peer event for member systemb
```

此普通的机密是实现堆栈成员SGBP出价和仲裁。

当远程客户端拨号到堆栈成员时，参考本文的Debugging PPP Multilink部分PPP链路验证讨论的。

一旦配线或路由问题，一个常见错误有(在SGBP hello消息实际上接收)的堆栈成员的源IP地址与同一个堆栈成员的本地定义的IP地址不同。

```
systema#debug sgbp error
%SGBP-7-DIFFERENT - systemb's addr 1.1.1.2 is different from hello's addr 3.3.4.5
```

这意味着SGBP从systemb接收的Hello的源IP地址不匹配为systemb配置的本地IP地址(通过**sgbp member命令**)。通过去systemb和检查更正此SGBP Hello能传送消息的多个接口。

错误的另一个常见原因是：

```
systema#debug sgbp error
%SGBP-7-DIFFERENT - systemb's addr 1.1.1.2 is different from hello's addr 3.3.4.5
```

这意味着您不安排systemk定义本地，但是另一个堆栈成员。

调试 PPP 多链路

检查的第一件事是否是在PPP和堆栈成员正确地验证的客户端。

因为是更加包含的，此示例展示CHAP认证。作为常见的例子，它使用Cisco平台，与本地用户名一起的一个客户端(加上(TACACS+)也支持终端访问控制器访问控制系统，但是没有显示此处)。

客户端用户x	**堆叠stackq每个成员**
#**config** username stackq password blah	#**config** username userx password blah

没有介入的拨号接口

因为没有在卸载服务器的拨号接口，需要虚拟访问接口*接口配置另一来源*。答案是虚拟模板接口。

1. 首先请确保多链路全局虚拟模板编号定义在每个堆栈成员。
   ```
   #config
   Multilink virtual-template 1
   ```
2. 如果未配置有问题的物理接口的任何拨号接口(例如PRI， BRI，异步和同步串行)，您可以定义：

```
interface virtual-template 1
ip unnumbered e0
ppp authen chap
ppp Multilink
```

注意： 您不定义了在虚拟模板的一个特定IP地址。这是因为设计的虚拟访问接口从虚拟模板接口总是被克隆。如果一随后的PPP链路也筹划对有活动的虚拟访问接口的一个堆栈成员已经被克隆和，您有在两个虚拟接口的相同IP地址，造成IP不正确路由在他们之间。

介入的拨号接口

当拨号程序在物理接口时配置，没有需要指定虚拟模板接口，因为接口配置位于拨号接口。在这种情况下，虚拟访问接口作为无源接口，支持在拨号接口和成员接口之间关联与拨号接口。

注意： 拨号接口，拨号1，在PPP多链路会话上显示如下：

```
systema#show ppp Multilink
Bundle userx 2 members, Master link is Virtual-Access4
Dialer interface is Dialer1
0 lost fragments, 0 reordered, 0 unassigned, 100/255 load
0 discarded,  0 lost received, sequence 40/66 rcvd/sent
members 2
Serial0:4
systemb:Virtual-Access6    (1.1.1.1)
```

LCP和NCP

所有成员接口的LCP状态必须是UP。IPCP、ATCP和其他NCP应该仅是UP在捆绑接口。

捆绑接口4 **show int**输出应该读如下：

```
router#show int  Virtual-Access4
Virtual-Access4 is up, line protocol is up
        :
    LCP Open, Multilink Open
    Open: ipcp
            :
```

其他成员接口应该有输出的以下**show int** ：

```
router# show int Serial0:4
Serial0:4 is up, line protocol is up
        :
LCP Open, Multilink Open
Closed:  ipcp
```

调试 VPN/L2F

打开以下：

```
debug vpn event
debug vpn error
```

当物理接口接受呼入呼叫和当前转发到目标栈成员时，您看到以下：

```
debug vpn event
debug vpn error
```

在目标栈成员上，如果看到以下：

```
debug vpn event
debug vpn error
```

然后请检查您的您的虚拟模板接口的定义。通常，虚拟模板接口必须匹配接受呼入呼叫物理接口的PPP接口参数。

切记最低配置(使用CHAP为例) ：

```
#config
multilink virtual template 4
int virtual-template 4
ip unnum e0
encap ppp
ppp authen chap
ppp Multilink
```

您可以发现以下：

```
#config
multilink virtual template 4
int virtual-template 4
ip unnum e0
encap ppp
ppp authen chap
ppp Multilink
```

如果看到上述消息，意味着L2F顺利地设想了从首先接纳呼入呼叫到堆栈成员同一个客户端的捆绑接口驻留的堆栈成员的PPP链路(或在卸载方案将创建，正如)。

常见错误是疏忽定义普通的堆叠名称(stackq)或不匹配的堆叠密码用户名在所有堆栈成员。

发出以下命令：

```
debug vpdn l2f-error
```

下列信息结果：

```
debug vpdn l2f-error
```

在这种情况下更正在每个堆栈成员的用户名和密码匹配。

相关信息

- 本文的第1部分
- 虚拟访问PPP特性在Cisco IOS软件方面
- 了解 VPDN
- 技术支持 - Cisco Systems