

WAN

本章将讨论 WAN（Wide-Area Networks，广域网）中最常用的两种数据链路层协议的相关细节信息：PPP（Point-to-Point Protocol，点对点协议）和以太网 WAN。

6.1 “我已经知道了吗？” 测试题

表 6-1 列出了本章的基本主题以及与之相应的测试题。

表 6-1 “我已经知道了吗？” 基本主题与测试题对照表

| 基本主题 | 测试题 |
|-------|-----|
| PPP | 1-3 |
| PPPoE | 4-5 |
| 城域以太网 | 6 |
| 得分 | |

为了提高每章前面的测试评价效果，请务必严格评分。参考答案请参见附录 A。

- 假设 PPP 链路出现故障且刚刚被重新恢复，那么最后协商的是下面哪一种特性？
 - CHAP 认证
 - RTP 头压缩
 - 环回链路检测
 - 链路质量监控
- 假设接口 s0/0、s0/1 和 s1/0 均处于 up 状态且均属于连接其他路由器的多链路 PPP 绑定，同时将多链路接口带宽设置为 1536。如果有一个 1500 字节的数据包被路由出该多链路接口，那么下面哪一项将确定该数据包从哪一条链路流出？
 - 当前 CEF FIB 以及 CEF 负载均衡方法
 - 当前快速交换缓存
 - 数据包将按照循环调度算法从某个接口发送出去
 - 这三条链路都会发送一个分段

210 第6章 WAN

3. R1 和 R2 通过一条专线连接, 并且连接接口均使用各自的 s0/1 接口。如果将 CHAP 配置为使用本地定义的用户名和密码, 那么下面关于这两台路由器的配置命令及配置模式的描述, 哪些是错误的?
- a. 接口子命令 **encapsulation ppp**
 - b. 接口子命令 **ppp authentication chap**
 - c. 在 R1 上配置全局命令 **username R1 password samepassword**
 - d. 在 R2 上配置接口子命令 **username R2 password samepassword**
4. 作为最低要求, 为了支持客户端的远程连接, 必须在 PPPoE 服务器上配置哪条命令?
- a. **ip dhcp**
 - b. **ip address**
 - c. **no virtual-reassembly**
 - d. **ip unnumbered**
 - e. **ip peer default ip address**
5. PPPoE 报头会给每个帧增加多少额外开销?
- a. 4 字节
 - b. 6 字节
 - c. 8 字节
 - d. 不增加开销
6. 下面哪一项架构需求不属于 VPLS 转发操作?
- a. 自动发现提供商边缘路由器
 - b. 用于连接 VSI 的伪线信令
 - c. 环路避免
 - d. 提供商边缘路由器的静态配置
 - e. MAC 地址撤销

基本主题

6.2 二层协议

点到点链路中最常见的两种二层协议就是 HDLC (High-Level Data Link Control, 高级数据链路控制) 和 PPP (Point-to-Point Protocol, 点对点协议)。由于 ISO 制定的较老的 HDLC 标准没有类型 (Type) 字段, 因而 Cisco 在 HDLC 实现中增加了 Cisco 专有的 2 字节类型字段, 以便在 HDLC 链路上支持多协议。Cisco 路由器将这种 Cisco

专有的 HDLC 版本作为其默认的串行接口封装方式。

6.2.1 HDLC



Cisco 路由器的默认串行接口封装方式是 Cisco HDLC，由于不需要在路由器上显式配置 Cisco HDLC，因而配置中也不会显示该封装类型。

对于背靠背的串行连接来说，连接电缆 DCE (Data Communications Equipment, 数据通信设备) 端的路由器为串行链路提供时钟信号。在接口配置模式下使用 **clockrate** 命令即可让电缆 DCE 端的路由器 (即本例中的 R1) 为串行链路提供时钟信号。运行 **show controllers** 命令即可确定电缆的哪一端连接在串行接口上。

从例 6-1 可以看出，电缆的 DCE 端连接至 R1，DTE (Data Terminal Equipment, 数据终端设备) 端连接至 R2，目前的时钟速率属于串行接口的默认配置，将自动应用于该串行链路。不过需要注意的是，大家可以根据不同的应用场景控制时钟速率，具体取决于设备所安装的环境。

例 6-1 HDLC 配置

```
! First we will apply the ip address to R1
! and "no shut" the interface
R1(config)# interface serial0/0
R1(config-if)# ip address 10.1.12.1 255.255.255.0
R1(config-if)# no shut
R1(config-if)#
! Now we will do the same for R2
R2# conf t
R2(config)# interface serial0/0
R2(config-if)# ip address 10.1.12.2 255.255.255.0
R2(config-if)# no shut
```

只要利用以下命令即可验证上述配置：

- **show controllers** 命令；
- **ping** 命令；
- **show interface** 命令。

例 6-2 的输出结果显示了这三条命令针对上述示例配置的运行结果。

例 6-2 验证 HDLC 配置

```
! Show controllers on R1:
R1# show controllers serial 0/0
Interface Serial0/0
Hardware is GT96K
DCE 530, clock rate 2000000
! ---- Output Omitted ----
!
! Now for the Ping test from R1 to R2:
!
R1# ping 10.1.12.2
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.12.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
R1#
!
! Lastly we will look at the output of the show interface serial 0/0 command on R1:
!
R1# show interface serial 0/0
Serial0/0 is up, line protocol is up
  Hardware is GT96K Serial
  Internet address is 10.1.12.1/24
  MTU 1500 bytes, BW 1544 Kbit/sec, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set
!
! Notice that the above output indicates that we are running HDLC Encapsulation
!
  Keepalive set (10 sec)
  CRC checking enabled
  Last input 00:00:01, output 00:00:02, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
    Conversations 0/1/256 (active/max active/max total)
    Reserved Conversations 0/0 (allocated/max allocated)
    Available Bandwidth 1158 kilobits/sec
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
    129 packets input, 8280 bytes, 0 no buffer
    Received 124 broadcasts, 0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    133 packets output, 8665 bytes, 0 underruns
    0 output errors, 0 collisions, 7 interface resets
    0 unknown protocol drops
    0 output buffer failures, 0 output buffers swapped out
    0 carrier transitions
  DCD=up DSR=up DTR=up RTS=up CTS=up
```

6.2.2 PPP

PPP 定义在 RFC 1661 中，包含一个协议（Protocol）字段和大量功能特性。表 6-2 列出了 HDLC 与 PPP 的主要对比信息。

表 6-2 HDLC 与 PPP 对比

| 功能特性 | HDLC | PPP |
|------------------|------|-----|
| 差错检测? | 是 | 是 |
| 差错恢复? | 否 | 是 |
| 标准的协议类型字段? | 否 | 是 |
| 是 IOS 串行链路的默认配置? | 是 | 否 |
| 支持同步和异步链路? | 否 | 是 |

*Cisco IOS 默认不使用该可靠的 PPP 特性，该特性允许 PPP 执行差错恢复功能

PPP 成帧 (RFC 1662) 使用简单的 HDLC 报头和报尾作为 PPP 帧格式的大部分字段 (如图 6-1 所示)。PPP 成帧只是在原始的 HDLC 成帧的基础上简单地增加了协议字段和可选的填充 (Padding) 字段 (填充字段允许 PPP 帧长度始终保持为偶数字节)。

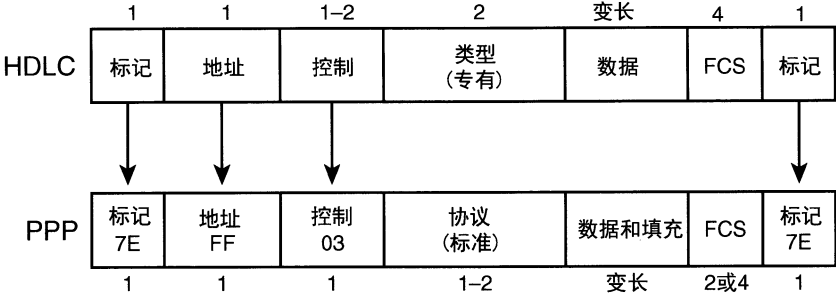


图 6-1 HDLC 与 PPP 成帧对比

1. PPP 链路控制协议

PPP 标准可以分为两大类,一类是与任何特定三层协议都无关的功能特性,另一类是与特定三层协议相关的功能特性。PPP LCP (Link Control Protocol, 链路控制协议) 负责控制与任何特定三层协议均无关的功能特性。对于 PPP 支持的每种三层协议来说,PPP 都会定义一个 NCP (Network Control Protocol, 网络控制协议)。例如,PPP IPCP (Internet Protocol Control Protocol, Internet 协议控制协议) 就为 IP 定义了相应的 PPP 功能特性,如动态地址分配。

PPP 串行链路刚刚启动时 (例如,路由器侦听到 CTS[Clear to Send, 清除发送]、DSR[Data Send Read, 数据发送读取]以及 DCD[Data Carrier Detect, 数据载波检测]之后就会在物理层启动),LCP 就开始与链路对端进行参数协商。例如,LCP 将控制协商采用何种认证方式、遵照何种认证顺序以及允许何种认证协议 (如 CHAP[Challenge Handshake Authentication Protocol, 挑战握手认证协议]) 来完成认证操作。如果所有的 LCP 协商过程均成功完成,那么 LCP 就处于 up 状态,此后 PPP 将开始启动三层控制协议。

表 6-3 列出了 LCP 的关键功能特性,后面将详细描述其中的一些关键 LCP 功能特性。

考试要点

表 6-3 PPP LCP 功能特性

| 功能特性 | 描述 |
|---------------------------------------|--|
| LQM (Link Quality Monitoring, 链路质量监控) | LCP 负责交换接收到的无错帧的比例统计信息,如果比例低于配置值,那么将丢弃该链路 |
| 环路检测 | 每台路由器都会生成并发送一个随机选择的魔术字,如果路由器收到了自己的魔术字,那么就表明该链路是环路并记录下来 |


```
*Apr 11 14:48:14.795: Se0/1/0 LCP: QualityType 0xC025 period 1000
(0x0408C025000003E8)
*Apr 11 14:48:14.795: Se0/1/0 LCP: MagicNumber 0x13403093 (0x050613403093)
*Apr 11 14:48:14.807: Se0/1/0 LCP: State is Open
! LCP completes, with authentication occurring next. In succession below, the
! challenge is issued in both directions ("O" means "output," "I" means "Input").
! Following that, the response is made, with the hashed value. Finally, the
! confirmation is sent ("success"). Note that by default the process occurs in
! both directions.
*Apr 11 14:48:14.807: Se0/1/0 PPP: Phase is AUTHENTICATING, by both
*Apr 11 14:48:14.807: Se0/1/0 CHAP: O CHALLENGE id 85 len 23 from "R3"
*Apr 11 14:48:14.811: Se0/1/0 CHAP: I CHALLENGE id 41 len 23 from "R4"
*Apr 11 14:48:14.811: Se0/1/0 CHAP: Using hostname from unknown source
*Apr 11 14:48:14.811: Se0/1/0 CHAP: Using password from AAA
*Apr 11 14:48:14.811: Se0/1/0 CHAP: O RESPONSE id 41 len 23 from "R3"
*Apr 11 14:48:14.815: Se0/1/0 CHAP: I RESPONSE id 85 len 23 from "R4"
*Apr 11 14:48:14.819: Se0/1/0 CHAP: O SUCCESS id 85 len 4
*Apr 11 14:48:14.823: Se0/1/0 CHAP: I SUCCESS id 41 len 4
*Apr 11 14:48:14.823: Se0/1/0 PPP: Phase is UP
```

3. 多链路 PPP

多链路 PPP (Multilink PPP) 可以缩写成 MPL、MP 或 MLPPP, 定义了一种在二层将多条并行的串行链路组合在一起的方法。MPL 的最初目标是组合多条不需要任何三层负载均衡特性的 ISDN B 通道, 但 MLP 可以为任何类型的点对点串行链路提供负载均衡功能。

MLP 实现流量均衡的方式是将每个数据链路层帧进行分段 (可以根据并行链路的数量或者根据已配置的分段时延进行分段), 然后再通过不同的链路发送这些分段。例如, 假设有三条并行链路, 那么 MLP 就会将每个帧都分为三个分段并通过这三条链路各发送一个分段。为了能够在接收端重组这些分段, MLP 需要为每个分段增加一个报头 (4 字节或 2 字节), 报头中包含一个序列号 (Sequence Number) 字段和标志 (Flag) 字段, 其中, 标志字段可以标明分段的起始和结束。

可以使用多链路接口或虚模板来配置 MLP。例 6-4 显示的 MLP 多链路接口拥有两个底层串行接口, 该配置示例显示了路由器 R4 向路由器 R3 经 MLP 连接发起的 ping 测试后的接口统计信息。

例 6-4 利用多链路接口配置 MLP 以及相应的统计信息——R3

```
! All Layer 3 parameters are configured on the multilink interface. The
! serial links are associated with the multilink interface using the ppp
! multilink group commands.
interface Multilink1
 ip address 10.1.34.3 255.255.255.0
 encapsulation ppp
 ppp multilink
 ppp multilink group 1
!
interface Serial0/1/0
 no ip address
```

```

encapsulation ppp
ppp multilink group 1
!
interface Serial0/1/1
no ip address
encapsulation ppp
ppp multilink group 1
! Below, the interface statistics reflect that each of the two serial links sends
! the same number of packets, one fragment of each original packet. Note that the
! multilink interface shows roughly the same number of packets, but the bit rate
! matches the sum of the bit rates on the two serial interfaces. These stats
! reflect the fact that the multilink interface shows prefragmentation
! counters, and the serial links show post-fragmentation counters.
R3# sh int s 0/1/0
Serial0/1/0 is up, line protocol is up
! lines omitted for brevity
  5 minute input rate 182000 bits/sec, 38 packets/sec
  5 minute output rate 182000 bits/sec, 38 packets/sec
    8979 packets input, 6804152 bytes, 0 no buffer
    8977 packets output, 6803230 bytes, 0 underruns
R3# sh int s 0/1/1
Serial0/1/1 is up, line protocol is up
! lines omitted for brevity
  5 minute input rate 183000 bits/sec, 38 packets/sec
  5 minute output rate 183000 bits/sec, 38 packets/sec
    9214 packets input, 7000706 bytes, 0 no buffer
    9213 packets output, 7000541 bytes, 0 underruns
R3# sh int multilink1
Multilink1 is up, line protocol is up
! lines omitted for brevity
  Hardware is multilink group interface
  Internet address is 10.1.34.3/24
  MTU 1500 bytes, BW 3088 Kbit, DLY 100000 usec,
    reliability 255/255, txload 31/255, rxload 30/255
  Encapsulation PPP, LCP Open, multilink Open
  Open: CDPCP, IPCP, loopback not set
  5 minute input rate 374000 bits/sec, 40 packets/sec
  5 minute output rate 377000 bits/sec, 40 packets/sec
    9385 packets input, 14112662 bytes, 0 no buffer
    9384 packets output, 14243723 bytes, 0 underruns

```

4. MLP 链路分段与交织

术语 LFI (Link Fragmentation and Interleaving, 链路分段与交织) 指的是一种 Cisco IOS QoS 工具, 该工具可以避免小的延迟敏感型数据包必须等待较大的延迟非敏感型数据包按串行方式从接口发送出去。为此, LFI 工具需要对较大的数据包进行分段, 只要发送了原始大数据包的部分分段之后就可以发送延迟敏感型数据包。LFI 的关键组件包括分段 (可以在不同数据包的分段之间进行交织的能力) 和排队调度器 (负责交织数据包)。从图 6-2 中可以看出, 1500 字节的数据包被分成了多个分段, 而且排队调度器将 60 字节的数据包交织到了前两个分段之后。

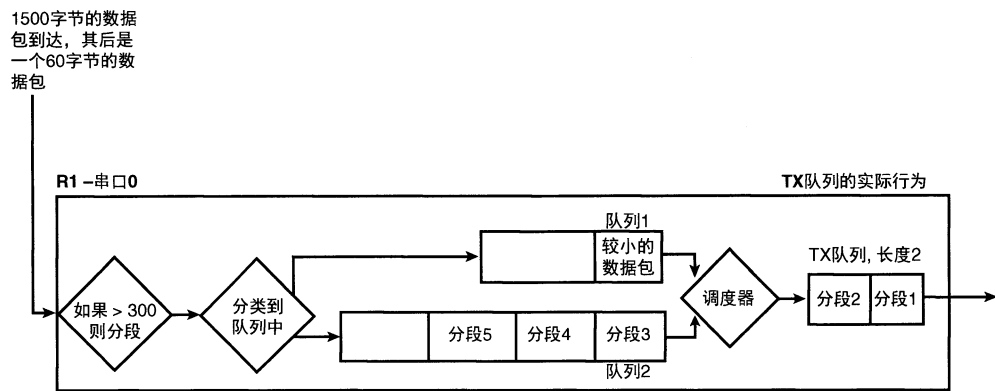


图 6-2 MLP LFI 的概念

MLP 支持 LFI，相应的关键信息如下：

考试要点

- 接口子命令 **ppp multilink interleave** 的作用是让路由器支持交织功能；
- 命令 **ppp multilink fragment-delay x** 的作用是间接定义分段大小（根据如下公式进行计算）。需要注意的是，由于延迟参数的单位是毫秒，因而必须对接口带宽的单位进行相应的转换：
分段大小= $x \times \text{带宽}$
- MLP LFI 可用于单条链路或多条链路；
- 多链路接口上的排队调度器负责确定下一个将要发送的数据包，因而很多实现都利用 LLQ，始终在分段之间交织延迟敏感型流量。

例 6-5 给出了例 6-2 的更新配置示例，例中启用了 LFI 特性。

例 6-5 利用 LLQ 交织语音流量的 MLP LFI

```
! The fragment delay is set to 10 ms, so the fragments will be of size (256,000 *
! .01 second) = 2560 bits = 320 bytes. The ppp multilink interleave command allows
! the queuing tool to interleave packets between fragments of other packets, and
! the referenced policy map happens to use LLQ to interleave voice packets.
interface Multilink1
 bandwidth 256
 ip address 10.1.34.3 255.255.255.0
 encapsulation ppp
 ppp multilink
 ppp multilink group 1
 ppp multilink fragment-delay 10
 ppp multilink8 interleave
 service-policy output queue-on-dscp
```

5. PPP 压缩

PPP 可以协商使用二层净荷压缩、TCP 头压缩以及/或 RTP 头压缩功能。每种类型的压缩功能都各有利弊，具体与所压缩的内容有关（如图 6-3 所示）。

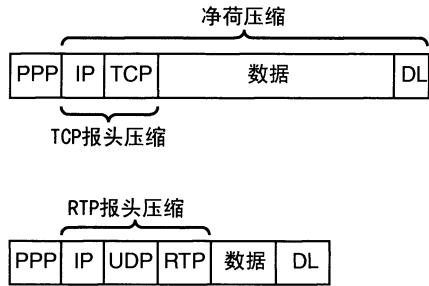


图 6-3 压缩功能所压缩的字段

净荷压缩与头压缩相比，净荷压缩适合长度较长的数据包，而头压缩则适合长度较短的数据包。头压缩利用报头的可预测性，报头字段的压缩率能够达到 10:1 至 20:1。但是，如果数据包内的数据长度远大于报头长度，那么头压缩所节省的字节数可能只占到总体带宽需求很小的一部分，此时净荷压缩更具吸引力。

6. PPP 二层净荷压缩

Cisco IOS 软件支持三种类型的 PPP 净荷压缩选项，分别为 LZS (Lempel-Ziv Stacker)、MPPPC (Microsoft Point-to-Point Compression, 微软点对点压缩) 和 Predictor。Stacker 和 MPCC 均使用相同的底层 LZ (Lempel-Ziv) 压缩算法，而 Predictor 则使用称为 Predictor 的压缩算法。与 Predictor 算法相比，LZ 使用的 CPU 更多但内存消耗较少，而且 LZ 通常能够获得更高的压缩比。

表 6-4 列出了净荷压缩的关键信息。需要注意的是，这三种压缩选项中只有 LSZ 同时支持帧中继和 HDLC 链路。此外，使用 ATM 到帧中继业务互通 (ATM-to-Frame Relay Service Interworking) 时，如果要启用净荷压缩功能，那么必须使用 MLP。因此，对于 PPP 所支持的所有净荷压缩类型来说，ATM 到帧中继业务互通也都支持。

表 6-4 PPP 净荷压缩工具：功能特性对比

| 功能特性 | Stacker | MPPC | Predictor |
|---------------------------------|---------|------|-----------|
| 使用 LZ 算法? | 是 | 是 | 否 |
| 使用 Predictor 算法? | 否 | 否 | 是 |
| HDLC 支持? | 是 | 否 | 否 |
| PPP 支持? | 是 | 是 | 是 |
| 帧中继支持? | 是 | 否 | 否 |
| 支持 ATM 以及 ATM 到帧中继业务互通 (使用 MLP) | 是 | 是 | 是 |

配置净荷压缩特性时，只要在链路两端的各个接口下配置相匹配的 **compress** 命令并为每种类型的压缩功能配置相匹配的参数即可。配置了压缩功能之后，PPP 就会启

动 CCP (Compression Control Protocol, 压缩控制协议, 属于另一种 NCP) 来执行压缩协商操作并管理相应的压缩进程。

7. 头压缩

PPP 支持两种类型的 IP 头压缩方式: TCP 头压缩和 RTP 头压缩 (图 6-3 给出了这两种压缩方式的报头示意图)。

语音流和视频流使用图 6-3 底部所示的 RTP 压缩。语音流 (特别是低比特率编解码器) 的数据字段非常短。例如, G.729 数据包通常是 60 字节, 60 字节中的 40 字节为 IP/UDP/RTP 报头, 而 RTP 头压缩能够将 IP/UDP/RTP 报头 (40 字节) 压缩为 2~4 字节, 因而在使用 G.729 的时候, RTP 头压缩能够将带宽需求降低 50% 以上。

TCP 头压缩可以将 IP 和 TCP 报头 (40 字节) 压缩为 3~5 字节, 因而对于净荷较小的 TCP 包来说, 带宽节省量非常可观, 计算结果与前面的 RTP 压缩示例相似。但是, 从 CPU 和内存消耗的角度来看, TCP 头压缩可能并不适用于长度较长的数据包。例如, 对于 1500 字节的数据包来说, 即使将报头从 40 字节压缩为 3 字节, 也只能将数据包大小降低 2% 左右。

配置头压缩功能时, 既可以使用两个传统的头压缩命令, 也可以使用 MQC (Modular QoS CLI, 模块化 QoS CLI) 命令。两个传统的头压缩命令分别为 **ip tcp header-compression [passive]** 和 **ip rtp header-compression [passive]**, 需要在链路每端的串行接口 (PPP) 或多链路接口 (MLP) 下配置这两条命令。配置完头压缩功能之后, PPP 就会使用 IPCP 来协商启用每种类型的头压缩功能 (如果配置了关键字 **passive**, 那么路由器就会等待其他路由器发起 IPCP 协商操作)。按照这种方式配置之后, 所有使用该链路的 TCP 流以及/或 RTP 流均会被压缩。

例 6-6 给出了另一种可选的配置方法, 即利用 MQC 策略映射来创建基于类别的头压缩。由于本例仅为类别为 Telnet 的流量应用了 TCP 头压缩功能, 因而将 TCP 头压缩应用到了最可能从 TCP 头压缩受益的数据包上, 而没有将 CPU 和内存资源浪费在压缩长数据包上 (由于 Telnet 仅在每个 TCP 报文段中发送一次击键信息, 因而除非配置了 **service nagle** 命令, 否则 Telnet 在默认情况下是非常低效的)。

例 6-6 MQC 基于类别的头压缩

```
! RTP compression is enabled in the voice class, TCP header compression in the
! critical data class, and no compression in the class-default class.
policy-map cb-compression
  class voice
    bandwidth 82
  compress header ip rtp
  class critical
    bandwidth 110
  compress header ip tcp
!
```

```
interface Multilink1
bandwidth 256
service-policy output cb-compression
```

6.2.3 PPPoE

PPPoE (PPP over Ethernet, 以太网上的 PPP) 可以通过简单的桥接设备将网络上的主机连接到远程汇集中器上。PPPoE 是宽带网络领域非常重要的一种接入协议, 通常是以软件栈的方式安装在终端客户 (用户) 的 PC 上。PPPoE 软件允许网络服务提供商 “拥有” 客户, 就像从客户 PC 向服务提供商网络运行的 PPP 会话一样。

PPPoE 通过共享介质 (通常是 DSL 服务提供商提供的宽带汇聚网络) 提供了一种仿真的 (且可选认证的) 点对点链路。事实上, 一种非常常见的应用场景就是在客户侧运行 PPPoE 客户端, 然后连接到 ISP 侧的 PPPoE 服务器并从服务器获得配置信息。

1. 服务器配置

例 6-7 给出了 ISP 侧必须实施的第一个配置步骤, 即创建一个 BBA (Broadband Aggregation, 宽带汇聚) 组以处理即将到来的 PPPoE 连接。为便于讨论, 我们将该 bba-group 命名为 **BBA-Group**, 并将其绑定到后面即将创建的虚模板上。

例6-7 虚模板配置

```
ISP(config)# bba-group pppoe BBA-GROUP
ISP(config-bba-group)# virtual-template 1
```

另外一个有用的配置选项就是部署 PPPoE 会话数限制, 从而作为一种保护手段。如果要限制基于关联 MAC 地址建立的 PPPoE 会话数, 那么就可以按下例输入相应的命令:

```
ISP(config-bba-group)# sessions per-mac limit 2
```

对于本应用场景来说, 我们将会话数限制为 2, 这样一来, 在前一个会话被丢弃或等待超时的情况下, 就能立即建立一个新会话。

接下来的一个重要步骤就是创建虚模板 (如前所述), 虚模板将为面向客户端的接口提供服务:

```
ISP(config)# interface virtual-template 1
```

如果 PPPoE 客户端向路由器发起了一个会话, 那么该路由器就会动态创建一个虚接口, 该接口将作为由该进程建立的点对点连接的入口。

为了保证正确操作, 必须为虚模板配置两个组件: IP 地址以及向客户端发布协商地址的 IP 地址池。该进程 (如例 6-8 所示) 是创建一个可运行的 PPPoE 进程的必备基础操作。

例6-8 PPPoE 基础配置

```
ISP(config-if)# ip address 10.0.0.1 255.255.255.0
ISP(config-if)# peer default ip address pool PPOE_POOL
```

接下来需要定义 IP 地址池的地址范围。如果要利用 IP 地址空间的起始地址和结束地址来配置名为 PPPOE_POOL 的本地 IP 地址池，那么就可以在全局配置模式下输入以下命令：

```
ISP(config)# ip local pool PPPOE_POOL 10.0.0.2 10.0.0.254
```

最后一步就是在面向客户端的接口上启用 PPPoE 组（如例 6-9 所示）。

例 6-9 PPPoE 启用 PPPoE 组

```
ISP(config)# interface f0/0
ISP(config-if)# no ip address
ISP(config-if)# pppoe enable group MyGroup
ISP(config-if)# no shutdown
```

尝试要点

请注意，该接口没有配置 IP 地址，本例将由前面配置的虚模板发布地址信息。

2. 客户端配置

与服务器配置相比，客户端的配置非常简单。首先创建一个拨号器接口来处理 PPPoE 连接，然后再与用来提供传输通道的物理接口进行关联。

创建 PPPoE 拨号器接口的配置方式如例 6-10 所示。

例 6-10 PPPoE 拨号器接口

```
CPE(config)# interface dialer1
CPE(config-if)# dialer pool 1
CPE(config-if)# encapsulation ppp
CPE(config-if)# ip address negotiated
```

尝试要点

命令 **ip address negotiated** 的作用是让客户端使用 PPPoE 服务器提供的 IP 地址。

作为该进程的一部分，PPP 报头会给每个帧增加 8 字节开销。如果数据包大小与默认的以太网 MTU（1500 字节）非常接近，那么该开销就有可能产生问题。为了解决这个问题，需要将指定拨号器接口的 MTU 减小到 1492，以免出现可能的分段问题。例 6-11 给出了相应的配置示例，同时还显示了将 ISP 接口分配给新创建的 PPPoE 拨号组的方式。

例 6-11 PPPoE 会话限制

```
CPE(config-if)# mtu 1492
CPE(config-if)# exit
CPE(config)# interface f0/0
CPE(config-if)# no ip address
CPE(config-if)# pppoe-client dial-pool-number 1
CPE(config-if)# no shutdown
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!If all is well, you should see a notification indicating the PPPoE session has
!!!successfully formed:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
%DIALER-6-BIND: Interface Vif bound to profile Df1
%LINK-3-UPDOWN: Interface Virtual-Access1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access1, changed
state to up
```



```

CPE(config)# interface dialer 1
CPE(config-if)# ppp chap password MyPassword
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!! We should see the PPPoE session come back up a few seconds later after
!!! successfully authenticating. debug ppp authentication can be used on the ISP
!!! router to monitor the CHAP exchange:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
ppp50 PPP: Using vpn set call direction
ppp50 PPP: Treating connection as a callin
ppp50 PPP: Session handle[E800003A] Session id[50]
ppp50 PPP: Authorization required
ppp50 CHAP: O CHALLENGE id 1 len 24 from "ISP"
ppp50 CHAP: I RESPONSE id 1 len 24 from "CPE"
ppp50 PPP: Sent CHAP LOGIN Request
ppp50 PPP: Received LOGIN Response PASS
Vi1.1 PPP: Sent LCP AUTHOR Request
Vi1.1 PPP: Sent IPCP AUTHOR Request
Vi1.1 LCP: Received AAA AUTHOR Response PASS
Vi1.1 IPCP: Received AAA AUTHOR Response PASS
Vi1.1 CHAP: O SUCCESS id 1 len 4

```

6.3 以太网 WAN

大家花时间研究一下以太网技术的各种功能及特性之后，就不会惊讶于业界已经成功地将以太网技术从常规的局域网部署环境推向了广域网部署环境。十年前，WAN 技术的协议及特性短名单中还包含帧中继、ATM、SONET（Synchronous Optical Network，同步光网络），但如今随着地理上相互隔离的网络环境之间的带宽需求越来越大，逐渐催生了一种新的网络互连解决方案。该解决方案利用以太网以及各种以太网特性作为骨干网，这些技术包括但不限于 VPLS（Virtual Private LAN Services，虚拟专用 LAN 服务）、MPLS（Multi-Protocol Label Switching，多协议标签交换）、ATOM（Any-Transport Over MPLS，MPLS 上的任意传输）、QnQ 隧道（Dot1Q-in-Dot1Q Tunnels 或 QnQ Tunnel）以及城域以太网（Metro-Ethernet）。

这些技术为我们创造了“无边界网络”，也就是可以在任何地方、任何时间提供服务的网络。此外，这些技术还提供了通常仅在总部机构才能实现的类似于 LAN 的用户性能，所有的这些都是通过 EWAN（Ethernet Wide-Area Networks，以太网广域网）技术来实现的。

以太网从最初的 LAN 技术逐步发展为适用于各种商业规模环境的可扩展的、性价比高的、可管理的 WAN 解决方案。与传统的 WAN 解决方案相比，以太网能够节约大量建设和运营成本，以太网 WAN 不但具有二层以太网的高带宽和简单性，而且扁平化的网络设计使得所连接站点看起来就像单个逻辑网络一样，大大简化了远程站点上连总部站点以及远程站点之间的互连问题。

为了便于讨论，下面将首先介绍 VPLS（VPLS 的设计目的是将二层连接传递到

EWAN 配置中的所有端点), 然后再介绍另一种被称为城域以太网的服务类型。人们通常将这两种技术简称为二层 VPN, 而且这两类技术通常都将 MPLS 作为使能技术。

6.3.1 VPLS

运营商可以利用 VPLS 技术将各种 WAN 连接 (IP 或 MPLS 网络上的 WAN 连接) 整合为单一的逻辑以太网, 从而可以在逻辑以太网中定义相应的 QoS 等级, 以确保语音和视频应用获得所需的带宽资源。

简而言之, VPLS 就是一种允许 MPLS 网络提供多点以太网 LAN 服务 (通常称为 TLS[Transparent LAN Service, 透明 LAN 服务]) 的体系架构。多点网络服务指的是允许 CE (Customer Edge, 客户边缘) 端点或节点与关联到该多点服务的其他所有 CE 节点进行直接通信的一种服务。与此相反, 使用点对点网络服务 (如 ATM) 时, 端用户通常要指定一个 CE 节点成为中心站点, 所有的分支站点都要连接到该中心站点上。如果某个分支站点需要与其他分支站点进行通信, 那么就必须通过中心节点。可以看出, 这种场景会引入额外的传输延迟。

为了提供多点以太网能力, IETF VPLS 草案提出了利用 MPLS PW (Pseudo-Wires, 伪线) 连接虚拟以太网网桥的方案。VPLS 在二层转发以太网帧的时候, 其操作特性与 IEEE 802.1 网桥完全相同, VPLS 可以自主学习源 MAC 地址到端口的关联关系, 并且基于目的 MAC 地址转发帧。如果目的 MAC 地址未知或者是广播地址或多播地址, 那么就会将该帧泛洪到与虚拟网桥相关联的所有端口。虽然 VPLS 的转发操作相对比较简单, 但 VPLS 的体系架构要求必须具备以下功能:

- 自动发现与特定 VPLS 实例相关联的其他 PE (Provider Edge, 提供商边缘) 设备;
- 支持 PW 信令以互连 VPLS VSI (Virtual Switch Instance, 虚拟交换实例);
- 避免环路;
- 撤销 MAC 地址。

6.3.2 城域以太网

虽然 MAN (Metropolitan-Area Network, 城域网) 环境中的以太网可以被用作纯以太网、MPLS 上的以太网 (Ethernet over MPLS) 或裸光纤上的以太网 (Ethernet over Dark Fiber), 但无论采用哪种传输介质, 我们都必须认识到在需要中等距离的骨干连接或城域 (同一个城市) 连接的网络部署方案中, 这种以太网 WAN 技术都是最佳方案。为什么会有这么多不同的城域以太网解决方案呢, 这是因为每种解决方案都有各自的优缺点。例如, 基于纯以太网部署方案成本较低, 但可靠性和扩展性较差, 而且通常仅限于小规模或实验部署环境。基于裸光纤的部署方案对于拥有现成基础设施的

场景来说很有用，而基于 MPLS 的部署方案虽然成本较高，但可靠性和扩展性很好，通常用于大型企业网络环境。

下面将简要讨论基于 MPLS 的城域以太网部署方案,这是因为 MPLS 技术是 CCIE R&S 笔试学习中的常见技术。基于 MPLS 的城域以太网在服务提供商网络中使用的是 MPLS，用户得到的以太网接口是铜线接口（如 100BASE-TX）或光纤接口（如 100BASE-FX）。客户的以太网数据包经 MPLS 网络进行传输，服务提供商的网络也将以太网作为底层技术来传输 MPLS。因此，基于 MPLS 的城域以太网实际上是 Ethernet over MPLS over Ethernet。

LDP（Label Distribution Protocol，标签分发协议）信令协议不但可以为内部标签（VC 标签）和 RSVP-TE（Resource Reservation Protocol-Traffic Engineering，资源预留协议-流量工程）提供站点到站点的信令功能，而且还可以为外层标签提供网络信令功能。

此外需要注意的是，典型的城域以太网系统都是星型网络或网状拓扑结构，路由器或服务器都是通过电缆或光纤介质进行互连的，这一点对于检测与排除城域以太网故障来说非常重要。

备考任务

表 6-5 列出了本章所涵盖的主要协议。

表 6-5 第 6 章的协议与标准

| 主题 | 标准 |
|---|----------|
| Point-to-Point Protocol (PPP) | RFC 1661 |
| PPP in HDLC-like Framing | RFC 1662 |
| PPP Internet Protocol Control Protocol (IPCP) | RFC 1332 |
| IP Header Compression over PPP | RFC 3544 |
| PPP Multilink Protocol (MLP) | RFC 1990 |
| PPPoE | RFC 2684 |
| Virtual Private LAN Service (VPLS) | RFC 4762 |
| Metro-Ethernet (GMPLS) | RFC 6004 |

表 6-6 列出了与本章讨论过的串行链路有关的常用 Cisco IOS 命令。

表 6-6 第 6 章的命令参考

| 命令 | 命令模式及描述 |
|---|---------------------------------------|
| <code>interface virtual-template number</code> | 全局模式；该命令可以为 MLP 创建虚模板接口并将用户移到虚模板配置模式中 |
| <code>ppp authentication { protocol1 [protocol2...] }</code> <code>[if-needed] [list-name default] [callin]</code> <code>[one-time] [optional]</code> | 接口模式；该命令可以定义认证协议（PAP、CHAP、EAP）以及其他参数 |
| <code>ppp multilink [bap]</code> | 接口模式；该命令可以在接口上启用 MLP |
| <code>ppp multilink fragment-delay delay-max</code> | 接口模式；该命令可以根据延迟及接口带宽定义分段大小 |
| <code>ppp multilink group group-number</code> | 接口模式；该命令可以将物理接口关联到多链路接口 |
| <code>ppp multilink interleave</code> | 接口模式；该命令允许在排队调度时将数据包交织到其他数据包的分段之间 |
| <code>compress [predictor stac mppc]</code> <code>[ignore-pfc]</code> | 接口模式；该命令可以配置净荷压缩 |
| <code>ip rtp header-compression [passive]</code> | 接口模式；该命令可以启用 RTP 头压缩 |
| <code>ip tcp header-compression [passive]</code> | 接口模式；该命令可以启用 TCP 头压缩 |
| <code>compression header ip [rtp tcp]</code> | 分类配置模式；该命令可以在 MQC 分类中启用 RTP 或 TCP 头压缩 |
| <code>ppp quality percentage</code> | 接口模式；该命令可以按照配置的百分比启用 LQM 监控特性 |
| <code>debug ppp negotiation</code> | 该命令可以启用调试功能，显示 PPP 协商的不同阶段 |

有关 `class-map`、`policy-map` 以及 `service-policy` 命令的详细信息，请参见第 5 章。

6.4 理解与记忆

与所有的 Cisco CCIE 笔试一样，CCIE 路由和交换笔试也包含了非常广泛的考试主题，因而本节提供了一些有用工具，来帮助大家加深理解并记忆本章所涵盖的考试主题。

6.4.1 凭记忆完成要点表

附录 E 包含了每章考试要点的摘要空表，请打印附录 E 并根据本章的表格信息，根据记忆填表，并参考附录 F 以作检查。

6.4.2 定义关键术语

请写出本章涉及的下列关键术语，然后在术语表中核对正确答案：

PPP、MLP、LCP、NCP、IPCP、CDPCP、PPPoE、LFI、CHAP、PAP、二层净荷压缩、TCP 头压缩、RTP 头压缩、VC、PVC、SVC、DTE、DCE、接入速率、接入链路、业务互通（使用 MLP）

6.4.3 推荐读物

www.cisco.com/c/en/us/td/docs/ios-xml/ios/bbds1/configuration/15-mt/bba-15-mt-book.html

<http://tools.ietf.org/html/draft-ietf-l2vpn-vpls-ldp-09>



本章主要讨论以下主题:

- IP 多播。