<http://blog.csdn.net/soloopin/article/details/8330419>

MTU: Maxitum Transmission Unit 最大传输单元

MSS: Maxitum Segment Size 最大分段大小

# **MTU**

最大传输单元（Maximum Transmission Unit，MTU）是指一种通信协议在某一层上面所能通过的最大数据报大小（以字节为单位），它通常与链路层协议有密切的关系。

EthernetII帧结构如下：



由于以太网传输电气方面的限制，每个以太网帧都有最小的大小64bytes，最大不能超过1518bytes，对于小于或者大于这个限制的以太网帧，我们都可以视之为错误的数据帧。一般的以太网转发设备会丢弃这些数据帧。（注：小于64Bytes的数据帧一般是由于以太网冲突产生的 “碎片”或者线路干扰或者坏的以太网接口产生的，对于大于1518Bytes的数据帧我们一般把它叫做Giant帧，这种一般是由于线路干扰或者坏的以太网口产生）。

由于以太网EthernetII最大的数据帧是1518Bytes，除去以太网帧的帧头（DMAC目的MAC地址48bit=6Bytes+SMAC源MAC地址48bit=6Bytes+Type域2bytes）14Bytes和帧尾CRC校验部分4Bytes （这个部份有时候大家也把它叫做FCS），那么剩下承载上层协议的地方也就是Data域最大就只能有1500Bytes，这个值我们就把它称之为MTU。

这个MTU就是网络层协议非常关心的地方，因为网络层协议比如IP协议会根据这个值来决定是否把上层传下来的数据进行分片。就好比一个盒子没法装下一大块面包，我们需要把面包切成片，装在多个盒子里面一样的道理。当两台远程PC互联的时候，它们的数据需要穿过很多的路由器和各种各样的网络媒介才能到达对端，网络中不同媒介的MTU各不相同，就好比一长段的水管，由不同粗细的水管组成（MTU不同 ）通过这段水管最大水量就要由中间最细的水管决定。

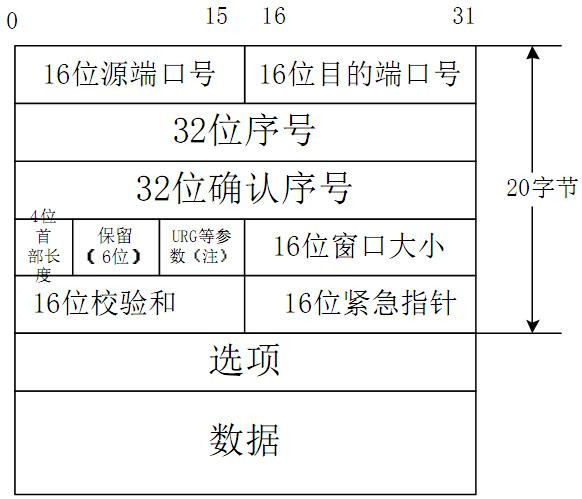
# **IP MTU**

对于网络层的上层协议而言（我们以TCP/IP协议族为例），网络层IP协议会检查每个从上层协议下来的数据包的大小，并根据本机MTU的大小决定是否作“分片”处理。分片最大的坏处就是降低了传输性能，本来一次可以搞定的事情，分成多次搞定，所以在网络层更高一层（就是传输层）的实现中往往会对此加以注意！有些高层因为某些原因就会要求我这个面包不能切片，我要完整地面包，所以会在IP数据包包头里面加上一个标签：DF（Donot Fragment）。这样当这个IP数据包在一大段网络（水管里面）传输的时候，如果遇到MTU小于IP数据包的情况，转发设备就会根据要求丢弃这个数据包，然后返回一个错误信息给发送者。这样往往会造成某些通讯上的问题，不过幸运的是大部分网络链路MTU都是等于1500或者大于1500。

对于UDP协议而言，这个协议本身是无连接的协议，对数据包的到达顺序以及是否正确到达不甚关心，所以一般UDP应用对分片没有特殊要求。对于TCP协议而言就不一样了，这个协议是面向连接的协议，对于TCP协议而言它非常在意数据包的到达顺序以及是否传输中有错误发生。所以有些TCP应用对分片有要求---不能分片（DF）。

# **TCP MSS**

MSS是最大传输大小的缩写，它是TCP协议里面的一个概念。如下图1-1所示：



在TCP报文中 MSS的位置就在选项的位置，根据RFC1323和RFC793规定，选项中内容有很多种，MSS是其中的一种，用kind=2表示；kind=1表示无操作，kind=4、5、6、7称为选择ACK及回显选项，但是由于回显选项已经被时间戳选项取代，同时，目前定义的选择ACK选项仍未定论，也没有包括在RFC1323中，所以具体代表什么含义还无定论。在实际网络数据传输，要求MSS+20TCP包头 +20 IP包头不大于MTU。MSS在TCP报文中是可选项，不是必选项，换句话说，MSS是可协商项，而且在协商过后，该选项内容可以改变，也可以没有；在协商MSS时，一般是建立TCP连结的两端发送Syn标志报文时互相通报，然后选取最小MSS作为双方的约定，如果双方都不通报或有一方不通报。

MSS就是TCP数据包每次能够传输的最大数据分段。为了达到最佳的传输效能，TCP协议在建立连接的时候通常要协商双方的MSS值，这个值TCP协议在实现的时候往往用MTU值代替（需要减去IP数据包包头的大小20Bytes和TCP数据段的包头20Bytes），所以往往MSS为1460。通讯双方会根据双方提供的MSS值得最小值确定为这次连接的最大MSS值。

注意：

ppp专网： 20 + 10 + 8 = 38

剩下payload： 1460 -38 = 1422

# **PMTUD**

<http://www.cisco.com/c/zh_cn/support/docs/ip/generic-routing-encapsulation-gre/25885-pmtud-ipfrag.html>

PMTU 发现过程：

对于一个基于网络的应用来讲，如果应用穿过网络的MTU与PMTU相等，那么应用穿过网络的效率最高，或者说，应用通过主机网卡发出的最大数据包与PMTU越接近(指小于等于PMTU)，应用穿过网络的效率越高，原因是有效的避免了分片和[重组](http://www.c114.net/keyword/%D6%D8%D7%E9)。

为了达到这个目的，一些操作系统支持自动发现路径MTU的功能，具体过程为：

路由器接口上收到一个报文长度大于本接口MTU值的报文，如果该报文被打上不分片的标记，将丢弃本报文，并且返回一个ICMP差错报文，通知报文发起者丢弃原因。报文发起者将发送比较小的报文。通过多次上述报文协商，将得到对于某一个固定路径上的最小Mtu值，这个过程叫做“Mtu Discovery”[详见RFC1191]。

“PMTU”发现存在的问题：

由于[互联网](http://www.c114.net/keyword/%BB%A5%C1%AA%CD%F8)上路由器或其它网络设备的配置的无法统一规范，某些[运营商](http://www.c114.net/keyword/%D4%CB%D3%AA%C9%CC)或网站考虑到[网络安全](http://www.c114.net/keyword/%CD%F8%C2%E7%B0%B2%C8%AB)和其它需要，有时会把ICMP报文过滤掉，此外，PMTU牵涉到主机、各类[交换机](http://www.c114.net/keyword/%BD%BB%BB%BB%BB%FA)、路由器、防火墙等网络设备，这些主机和网络设备没有有效的手段实现PMTU的协商和交互,这样Mtu Discovery不能正常运行，影响应用正常运行，即实质上目前没有有效的手段来发现PMTU。

互联网上的网络设备,遇到MTU发现报文或必须将IP包分片但DF设置为1时，路由器可采用以下任一种方式(从网上摘录)：

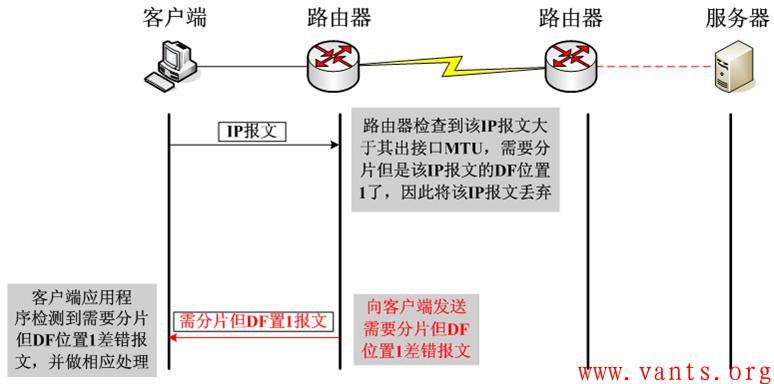
发送符合 RFC 792 中最初定义的“ICMP Destination Unreachable-Fragmentation Needed and DF Set”消息，然后丢弃该包。原始消息格式中不包含有关转发失败的链路的 IP MTU 的信息。（导致PMTU无法正常发现）  
⎫ 发送符合 RFC 1191 中重新定义的“ICMP Destination Unreachable-Fragmentation Needed and DF Set”消息，然后丢弃该包。此新消息格式包含一个 MTU 字段，可指出转发失败的链路的 IP MTU。（PMTU可能会正常发现）

RFC 1191 定义了路径 MTU (PMTU) 发现，它使得源和目的 TCP 对等方能够动态地发现二者之间路径的IP MTU，从而发现该路径的 TCP MSS。一旦收到符合 RFC 1191 定义的“Destination Unreachable-Fragmentation Needed and DF Set”消息，TCP 就会将该连接的 MSS 调整为指定 IP MTU 减去 TCP 和 IP 报头的大小。这样，在该 TCP 连接上发送的后续包就不会超过最大大小，无需分段即可在该路径上传输。

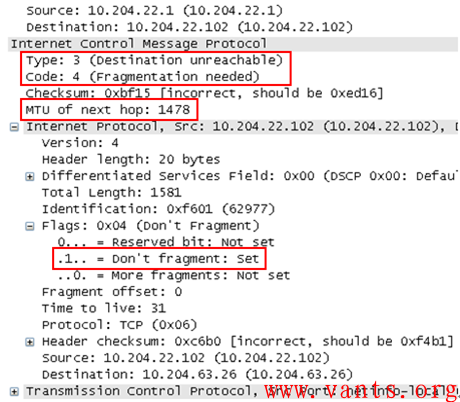
直接丢弃包。直接丢弃需分段但 DF 标记设置为 1 的包的路由器称为 PMTU 黑洞路由器。

总之，PMTU的不可发现性，导致因MTU问题引起的应用系统无法正常运行情况时有发生。

一旦DF位置一，将不允许中间设备对该报文进行分片，那么在遇到IP报文长度超过中间设备转发接口的MTU值时，该IP报文将会被中间设备丢弃。在丢弃之后，中间设备会向发送方发送ICMP差错报文。  
为了简单直观的展示这个交互的过程，我做了下面这个图示：



我找了一个实际环境下捕获的ICMP需要分片但DF位置一的差错报文，下图为其解码格式：

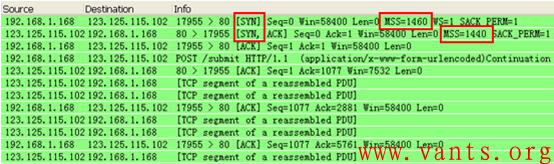


    我们可以看到其差错类型为3，代码为4，并且告知了下一跳的MTU值为1478。在ICMP差错报文里封装导致此差错的原始IP报文的报头（包含IP报头和四层报头）。

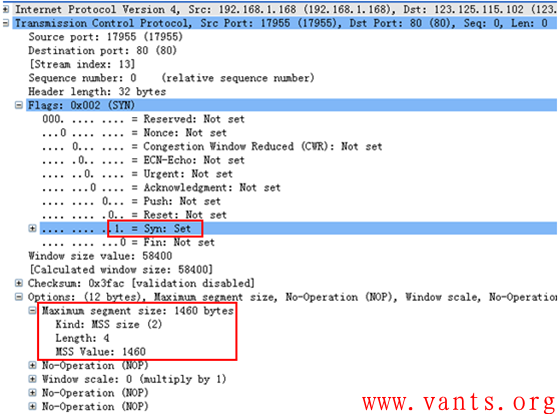
       一旦出现这种因DF位置一而引起丢包，如果客户端无法正常处理的话，将会导致业务应用出现异常，外在表现为页面无法打开、页面打开不全、某些大文件无法传输等等，这将严重影响业务的正常运行。  
那么客户端如何处理这种状况呢？  
TCP主要通过两种方式来应对：  
**1， 协商MSS，在交互之前避免分片的产生  
2， 路径MTU发现（PMTUD）**

**TCP MSS**

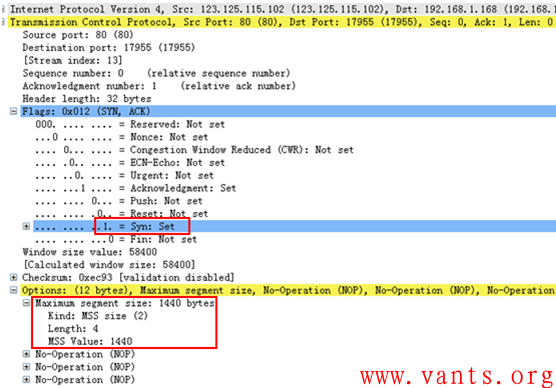
TCP在三次握手建立连接过程中，会在SYN报文中使用MSS（Maximum Segment Size）选项功能，协商交互双方能够接收的最大段长MSS值。  
 MSS是传输层TCP协议范畴内的概念，顾名思义，其标识TCP能够承载的最大的应用数据段长度，因此，MSS=MTU-20字节TCP报头-20字节IP报头，那么在以太网环境下，MSS值一般就是1500-20-20=1460字节。  
客户端与服务器端分别根据自己发包接口的MTU值计算出相应MSS值，并通过SYN报文告知对方，我们还是通过一个实际环境中捕获的数据报文来看一下MSS协商的过程：

[](http://www.vants.org/content/plugins/kl_album/upload/201209/13f465a71d39e31b28c1568269e577f3201209052157101372631015.png)

       这是整个报文交互过程的截图，我们再来看一下客户端的报文详细解码：

[](http://www.vants.org/content/plugins/kl_album/upload/201209/6cd05c27bad36f1e70485d9063b65c0520120905215711916817182.png)

       上图为客户端的SYN报文，在其TCP选项字段，我们可以看到其通告的MSS值为1460；我们在看看服务器端的SYN/ACK报文解码：

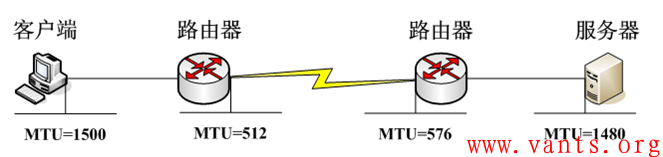
[](http://www.vants.org/content/plugins/kl_album/upload/201209/4150fa5a19d47041aa9727148deca512201209052157112105531868.png)

       上图为服务器端给客户端回应的SYN/ACK报文，查看其TCP选项字段，我们可以发现其通告的MSS值为1440。

       交互双方会以双方通告的MSS值中取最小值作为发送报文的最大段长。在此TCP连接后续的交互过程中，我们可以清楚的看到服务器端向客户端发送的报文中，TCP的最大段长度都是1440字节，如下图解码所示：

[](http://www.vants.org/content/plugins/kl_album/upload/201209/6f75c843874a0009964b5e24b2f24f57201209052157111939739669.png)

       通过在TCP连接之初，协商MSS值巧妙的解决了避免端系统分片的问题，但是在复杂的实际网络环境下，影响到IP报文分片的并不仅仅是发送方和接收方，还有路由器、防火墙等中间系统，假设在下图的网络环境下：

[](http://www.vants.org/content/plugins/kl_album/upload/201209/6fd282222635b1845ec97902a99baa83201209052157122094684129.png)

       中间路径上的MTU问题，端系统并不知道，因此需要一个告知的机制，这个机制就是路径MTU发现（PMTUD： Path MTU Discovery ）！

**PMTUD**

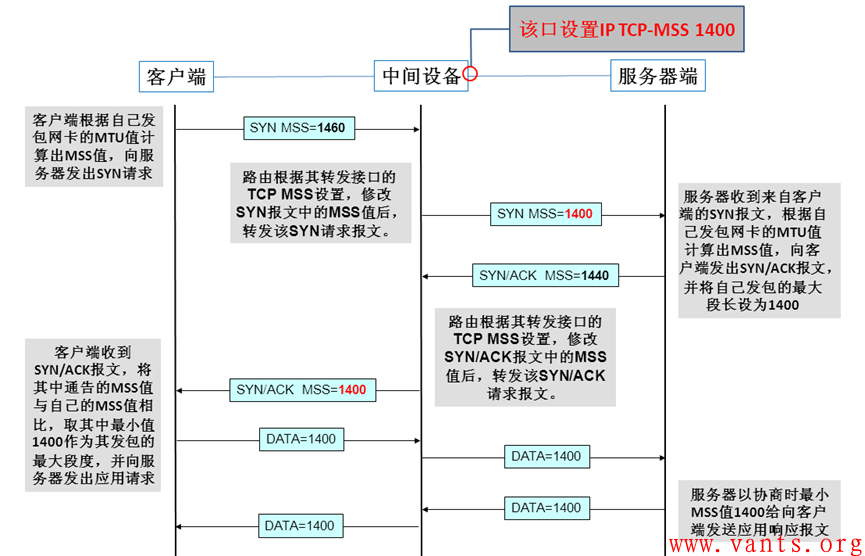
       说起PMTUD，我们必须在此回到上面讲到的ICMP需要分片但DF位置一差错报文，还记得那个ICMP差错报文中有一个字段是告知下一跳的MTU值的吗？PMTUD正是利用ICMP需要分片但DF位置一差错报文的这一特性来实现的。  
       发送方在接收到该差错报文后，会根据该报文给出的下一跳的MTU值计算适合传输的最大段长度，从而在后续的发送报文过程中，避免在中间路径被分片的情况产生。  
       这在端系统主要是通过在路由表里临时添加目的主机路由并将ICMP差错报文告知的下一跳MTU值跟该主机路由关联起来来实现。  
       PMTUD的确是个非常不错的机制，但是在复杂的实际网络环境中，有时候会失效，因为为了安全起见，有些网络管理员会在路由器、防火墙等中间设备上设置过滤ICMP报文的安全策略，这将导致ICMP差错报文被这些中间设备丢弃，无法达到发送方，从而引起PMTUD的失效，网上有个宫一鸣前辈共享的案例——《错误的网络访问控制策略导致PMTUD 实现故障一例》，该案例正是说明这种情况绝好的例子，大家可以自行百度此文档学习参考。

       值得一提的是PMTUD仅TCP支持，UDP并不支持PMTUD。

       由于PMTUD可能存在ICMP差错报文被过滤的情况，很多中间设备的接口支持adjust tcp mss设置功能，思科路由器一般是在接口模式下使用命令“ip tcp adjust-mss 1400 ”来做设置，其他的品牌产品的相关设置大家可在实际工作环境下自查相关品牌和产品的使用手册。

        这个功能主要是通过由中间设备修改经过其转发的TCP SYN报文中的MSS值，让中间设备参与进TCP 三次握手时SYN报文的MSS协商来避免分片。

       需要注意的是，该功能不像MTU值，只针对出接口，此功能一旦开启，其将针对该接口的收发双向有效。   
我做一个简化环境下的工作过程图示以便于大家理解其工作过程：

[](http://www.vants.org/content/plugins/kl_album/upload/201209/2b201320d835ec05a74f64202c0f2d7a2012090521570819765176.png)

# **区别及联系**

由前面的叙述可知：

MTU是一个二层的概念，以太网最大的mtu就是1500（它是不包含二层头部的，加上头部应该为1518 bytes），当然这里说的是很常规的情况，也有些server，比如server 2008，出来的就是jumbo frame了，我们在这里讨论常规情况。

IP MTU是一个三层概念，它包含了三层头部及所有载荷，根据下层为上层服务的，上层基于下层才能做进一步的扩展的原则，尽管IP MTU的变化范围很大（68-65535），但也不得不照顾以太网MTU的限制,说白了就是ip对以太网的妥协。

MSS是TCP里面的一个概念，它是TCP数据包每次能够传输的最大数据分段，不包含包头部分，它与IP MTU满足如下关系：IP MTU=MSS+20bytes（IP包头）+20bytes（TCP包头）。

当然，如果传输的时候还承载有其他协议，还要加些包头在前面，简言之，mtu就是总的最后发出去的报文大小，MSS就是需要发出去的数据大小，比如PPPoE，就是在以太网上承载PPP协议（点到点连接协议），它包括6bytes的PPPoE头部和2bytes的PPP协议ID号，此时，由于以太网的MTU值为1500，所以上层PPP负载数据不能超过1492字节，也就是相当于在PPPOE环境下的MTU是1492字节，MSS是1452字节。

如果有报文载 荷（不包括以太网封装，一般是数据大小加上协议头部，比如PING1472大小的报文，则载荷部分为1472 数据大小+20 IP头＋8 ICMP头＝1500字节，下同）大于MTU，那么这个报文在传送时就会被分片。

MRU（最大接收报文长度） 是PPP里的术语，它是数据链路层的一个参数，是在 PPP链路建立的时候协商的，如果PPP链路一端设备需要发送的报文载荷大于对端的 MRU，这个报文在传送时就会被分片；MSS 是TCP里的术语，它是网络层的参数，是 在TCP联接建立时协商的，在使用TCP传送数据时，如果数据长度大于MSS的值，数据就会被分成几个报文发送，注意：这和报文分片是不一样的，报文分片后只有第一个报文包含有报文详细的协议和端口等信息，而后面的每片只包含有IP头部及数据，和第一个报文有相同的报文ID。

传统以太网端口的MRU是1518.现在应该早就超过这个限制。

**PPPoE: PPP Over Ethernet（在以太网上承载PPP协议）**  
　　花开两朵，各表一枝，说完MTU的故事我们该讲讲今天的第二个主角—PPPoE所谓PPPoE就是在以太网上面跑PPP协议，有人奇怪了，PPP协议和Ethernet不都是链路层协议吗？怎么一个链路层跑到另外一个链路层上面去了，难道升级成网络层协议了不成。其实这是个误区：就是某层协议只能承载更上一层协议。为什么会产生这种奇怪的需求呢？这是因为随着宽带接入（这种宽带接入一般为Cable Modem或者xDSL或者以太网的接入）由于以太网缺乏认证计费机制而传统运营商是通过PPP协议来对拨号等接入服务进行认证计费的，所以就出了这么一个怪胎：PPPoE。（有关PPPoE的详细介绍参见V大以及本站其他成员的一些介绍文章，我就不啰里啰唆的了）。

PPPoE带来了好处，也带来了一些坏处，比如：二次封装耗费资源，降低了传输效能等等，这些坏处俺也不多说了，最大的坏处就是PPPoE导致MTU变小了以太网的MTU是1500，再减去PPP的包头包尾的开销（8Bytes），就变成1492。如果两台主机之间的某段网络使用了PPPoE那么就会导致某些不能分片的应用无法通讯。这个时候就需要我们调整一下主机的MTU，通过降低主机的MTU，这样我们就能够顺利地进行通讯了。

# **MTU问题解决方法**

通常情况下，MTU不匹配会表现为两种故障情况：

1. ping大包时不通
2. 无法访问某些站点

在这种情况下，通常有两种解决方法：

1. 修改用户端MTU值（不推荐使用）
2. 修改传输路由所有设备MTU值，确保路径MTU值大于用户发送的IP报文的长度，以保证用户报文不会因为超过设备的MTU值被丢弃。主要要考虑下面几种情况：

· 对于纯IP网络，要保证：路径MTU值>最大用户报文长度

· 对于纯MPLS网络（没有VPN业务），要保证路径MTU值>最大用户报文＋一层标签长度（4）

· 对于三层VPN业务，要保证：路径MTU值>最大用户报文＋两层标签长度（8）；

· 对于二层VPN业务，要保证：路由MTU值>最大用户报文长度＋两处标签长度（8）＋二层帧头长度（18）。

值得注意的是：fastethernet接口不能调整MTU，所以说在有些设备中，使用MTU命令不能解决问题的。此外，更改MTU后，如果IGP是OSPF的话，不同的MTU可能会造成OSPF 停留在INIT状态，此时需要将两端的MTU调整一致。

# **案例分析1**

下面通过一个某公司TCP MSS配置不匹配导致应用不正常的案例分析，来加深对上述概念的理解。其拓扑图如下图1-2所示:

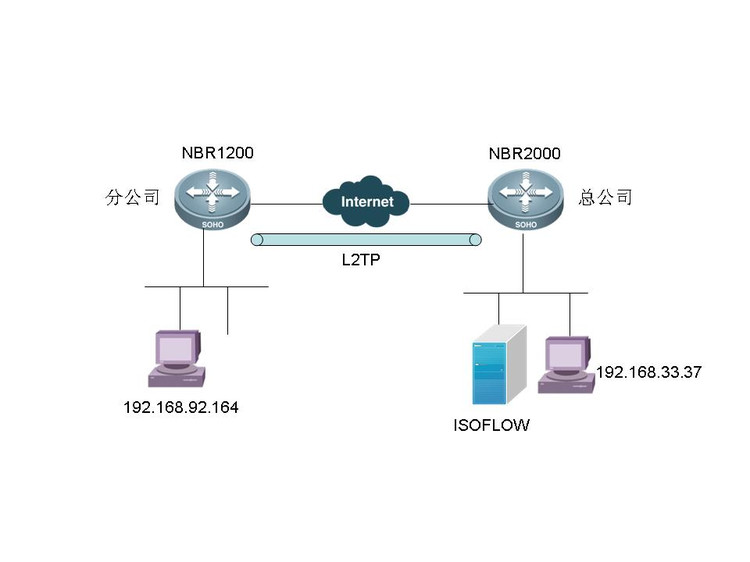


图1-2 某公司网络拓扑示意图

|  |  |  |
| --- | --- | --- |
| 设备 | 角色 | 说明 |
| NBR2000 | 公司总出口路由器 |    NBR2000启用L2TP功能，同时在L2TP虚拟模板接口中配置了TCP MSS为1300。     NBR1200启用2LTP功能，同时在L2TP虚拟模板接口中配置了TCP MSS为1400。     NBR1200与NBR2000通过L2TP协议建立隧道，NBR2000作为服务端，NBR1200作为客户端。 |
| NBR1200 | 分公司出口路由器 |

**故障现象：**

   在分公司NBR1200的PC上登录总公司ISOFLOW服务器，在ISOFLOW里签核表单的时候，会无法签核下去而导致IE死掉的现象。 如果碰见ISOFLOW单子的附件，会发现附件要过很长、很长的时间才能打开。

   访问ERP系统普通表单的速度不错，但是一旦进行数据库的读写，速度极慢；数据查询时经常超时。

1.故障诊断的思路

登录到总公司NBR2000和分公司NBR1200，查看配置，具体如下：

NBR2000 虚拟模板配置：

interface Virtual-Template 1

mtu 1400

ppp authentication pap

ip nat inside

**ip tcp adjust-mss 1300**

ip unnumbered GigabitEthernet 0/0

peer default ip address pool VPDN

NBR1200 虚拟模板配置：

interface Virtual-ppp 1

pseudowire X.X.X.X 20 encapsulation l2tpv2 pw-class pw

mtu 1400

ppp pap sent-username star-net password 7 0058274d4c24

**ip tcp adjust-mss 1400**

ip address negotiate

故障诊断的分析

1、  判断网络层是否正常

在现场从分公司PC PING总公司ISOFLOW服务器IP，长PING没有丢包，说明网络层是正常的。

2、应用异常

目前用户应用是通过HTTP来访问，出现能访问，但很慢的现象，说明是在应用层上出现了问题。一般来说，在应用层出问题，在排除软件本身问题的情况下，基本可以定位是MTU或TCP MSS不匹配原因引起。

3、TCP MSS分析

一般来说一般的应用软件，当客户端和服务器端在建立TCP连接的时候需要根据实际传输的报文大小来协商TCP的窗口大小MSS。在TCP协商的第一个报文也就是SYN置位的报文中会告知对方自己所能接收的最大TCP数据净荷（也就是TCP报文中的数据长度）----MSS，对方收到TCP同步报文也会回复一个SYN置位的TCP报文，其中也携带了MSS参数来告知对方自己能接收的最大MSS。根据以上分析，我们来看一下总公司与分公司路由器是怎么协商TCP MSS的：

（1）分公司-》总公司

PC发起TCP SYN报文与总公司ISOFLOW服务器进行协商，在TCP SYN报文经过分公司路由器虚拟模板接口（out方向）时，路由器会将接口配置的MSS值1400（分公司路由器在虚拟模板接口上配置了**ip tcp adjust-mss 1400**）与TCP报文中的MSS值1460（windows XP系统默认是1460）比较，取二者的最小值1400。ISOFLOW服务器收到此TCP同步报文，查看其中的MSS=1400，服务端在后续发送数据时封装的最大TCP数据载荷为1400。一般来说TCP MSS是不分包的，DF置位为1。

（2）总公司-》分公司

ISOFLOW发响应TCP SYN ACK报文给分公司的PC，在TCP SYN报文经过总公司路由器虚拟模板接口（out方向）时，路由器会将接口配置的MSS值1300（总公司路由器在虚拟模板接口上配置了**ip tcp adjust-mss 1300**）与TCP报文中的MSS值1460（windows XP系统默认是1460）比较，取二者的最小值1300。PC收到此TCP同步确认报文，查看其中的MSS=1300，PC在后续发送数据时封装的最大TCP数据载荷为1300。一般来说TCP MSS是不分包的，DF置位为1。

PC在向ISOFLOW发送请求，作为ISOFLWO的被请求端，发出的数据包括的数据文件，数据包较大，需要封装成最大MSS载荷传送（提高效率），在此MSS=1400,当数据包经过总公司L2TP VPN虚拟模板接口时，由于配置了**mtu 1400**，而此时数据包的MTU（MTU=1400+TCP(20)+IP(20)+PPP (4)+L2TP（8）+UDP（8）+IP（20）=1480）远远大于总公司路由器虚拟模板接口MTU 1400，且IP包头强制不分片位被置位，导致在出VPN隧道时报文被丢弃，所以现象就是能够建立TCP连接，但服务器端返回大数据包时，出现IE应用程序没有响应现象。同理，如果是从服务端发起的访问客户端口PC，由于客户端接收到的TCP同步报文中的MSS=1300，那么其封装的数据经过分公司路由器L2TP VPN隧道时，数据包的MTU(MTU=1300+ TCP(20)+IP(20)+PPP (4)+L2TP（8）+UDP（8）+IP（20）=1380)小于分公司路由器隧道MTU=1400，所以能够正常。

一般以太网MTU都为1500, 所以在以太网中, 往往TCP MSS为1460。

当一个连接建立时，连接的双方都要通告各自的M S S。它是一个可“协商”选项，当建立一个连接时，每一方都有用于通告它期望接收的M S S选项（M S S选项只能出现在S Y N报文段中）。一般说来，如果没有分段发生， M S S还是越大越好。报文段越大允许每个报文段传送的数据就越多，相对I P和T C P首部有更高的网络利用率。当T C P发送一个S Y N时，或者是因为一个本地应用进程想发起一个连接，或者是因为另一端的主机收到了一个连接请求，它能将M S S值设置为外出接口上的M T U长度减去固定的I P首部和T C P首部长度。对于一个以太网， M S S值可达1 4 6 0字节。

协商TCP MSS大小具体过程如下：  
TCP client发出SYN报文，其中option选项填充的MSS字段一般为(MTU-IP头大小-TCP头大小)，同样TCP server收到SYN报文后，会发送SYN＋ACK报文应答，option选项填充的mss字段也为(MTU-IP头大小-TCP头大小)；协商双方会比较SYN和SYN+ACK报文中MSS字段大小，选择较小的MSS作为发送TCP分片的大小。

对于涉及PPPOE＋NAT、IPsec、L2TP、GRE等组网，通常由于报文太大需要分片，这样会降低传输速率; 所以选择一个合适的MSS对传输数据来说比较重要. [**Linux**](http://lib.csdn.net/base/linux)中一般可以通过netfilter iptables设置TCP MSS来解决。

**iptables -A FORWARD -p tcp- -tcp-flags SYN,RST SYN -j TCPMSS --clamp-mss-to-pmtu**

这条规则的目的就是改变TCP MSS以适应PMTU(Path MTU)

**iptables -A FORWARD -p tcp --tcp-flags SYN,RST SYN- j TCPMSS --set-mss 128**

设置MSS为128

以下是一段内核中修改TCP MSS的代码:

static inline u32 set\_tcp\_mss(struct sk\_buff \*pskb, struct tcphdr \*tcph, u16 mtu)

{

u32 optlen, i;

u8 \*op;

u16 newmss, oldmss;

u8 \*mss;

if ( !tcph->syn )

return 0;

// 判断是否为合法tcp选项

if (tcph->doff\*4 < sizeof(struct tcphdr))

return 0;

optlen = tcph->doff\*4 - sizeof(struct tcphdr);

if (!optlen)

return 0;

// 扫描是否有MSS选项

op = ((u8\*)tcph + sizeof(struct tcphdr));

for (i = 0; i < optlen; ) {

if (op[i] == TCPOPT\_MSS

&& (optlen - i) >= TCPOLEN\_MSS

&& op[i+1] == TCPOLEN\_MSS) {

u16 mssval;

//newmss = htons( 1356 );

oldmss = (op[i+3] << 8) | op[i+2];

mssval = (op[i+2] << 8) | op[i+3];

// 是否小于MTU-( iphdr + tcphdr )

if ( mssval > mtu - 40 ) {

newmss = htons( mtu - 52 );

}

else {

break;

}

//

mss = &newmss;

op[i+2] = newmss & 0xFF;

op[i+3] = (newmss & 0xFF00) >> 8;

// 计算checksum

inet\_proto\_csum\_replace2( &tcph->check, pskb,

oldmss, newmss, 0);

mssval = (op[i+2] << 8) | op[i+3];

dprintf( "Change TCP MSS %d to %d/n", ntohs( oldmss ), mssval );

break;

}

if (op[i] < 2)

i++;

else

i += op[i+1] ? : 1;

}

return 0;

}

# **案例分析2**

ABC三地，A华为100E，BC各是华为的AR2811，作了gre vpn，ping值稳定在10MS。但就是打开内网erp软件的时候非常慢。这软件是web型式的，当时那俩从北京请来的工程师想了半天，加了条“tcp mss 1024”，问题解决。  
这是一年前的事了，前几天又收购了一家子公司D，想把这家子公司也加到这个vpn里面。摧的比较急，市场上找不到ar2811了，只好找了台USG2160。下面让哥头痛了6天的麻烦事就开始了。  
第一天，从A 地任意pc机ping   D地pc可以通。 从2160上ping  A地任意也可以通。就是2160下任何pc机ping不通A地。  
第二天，上级供货商联系工程师调，没结果。没办法打400找了华为工程师，华为在他们的实验室调试的结果是可以通的，让我们查查局域网是不是有毛病。  
第三天，经过哥推理吧，局域网没问题。再找华为工程师，这时换了另一位工程师，他查了半天，把  
acl number 3000  
rule 5 permit ip source 192.168.4.208 0.0.0.15  
改写成  
acl number 3000  
rule 0 deny ip source 192.168.4.0 0.0.0.255 destination 192.168.0.0 0.0.0.255  
rule 5 permit ip source 192.168.4.208 0.0.0.15

这时vpn通了，ping值平均8ms。本来以为大功告成，可以收钱了。顺手试了下那该死的erp软件，他良的，密码验证不过去.........  
第四天，联系软件商，逼着他们查了半天，结果是软件OK。华为那边也一直在想办法，晚上华为在D地pc机上抓包。  
第五天，问题转到了北京华为专门研发usg产品部门，根据抓包分析，加了条  
firewall tcp-mss 1024

不通，再加  
firewall defend syn-flood enable  
firewall defend syn-flood interface Vlanif1 tcp-proxy on  
谢天谢地，那该死的软件终于可以登录了。  
这个问题比较特殊，可能碰到的少。我想还是跟大家分享一下，W一哪天用得到呢。  
另外就是想问问，红色的几条命令都是啥意思，小弟嘛都不懂。  
写作水平太低，不知道表达清楚没有，大家将就着看吧。

################################################################

3月2日,vpn已通，ping值还算稳定，华为工程师说是这条命令的问题  
acl number 3000  
rule 5 permit ip source 192.168.4.208 0.0.0.15  
他加了一条命令  
acl number 3000  
rule 0 deny ip source 192.168.4.0 0.0.0.255 destination 192.168.0.0 0.0.0.255  
rule 5 permit ip source 192.168.4.208 0.0.0.15  
这vpn是通了，可是又有新的问题。  
在A地防火墙100E下面有台服务器192.168.0.200，有erp软件和一个类似于QQ的聊天工具。erp软件是web界面登陆的，发现不能验证密码，erp界面直接像是打不开网页的样子一直在走进度条，类似QQ的聊天软件提示密码错误或者连接服务器错误。但是用外网地址登陆erp软件的话就没问题。问过erp软件商，他们说是不应该是软件设置的问题。麻烦各位大锅再想想是不是防火墙哪里拦截了？  
  
  
  
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&以前的问题&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&  
小弟又有问题麻烦各位。  
有ＡＢＣＤ三个地方，A地shengji 用华为的100E防火墙，BC两块华为ＡＲ２８１１路由器，vpn调通正常使用已快一年。现在多了一个D yinxin，华为USG2610+华为S3328.调试后具体情况如下：  
A地192.168.0.X  
B地192.168.1.1  
C地192.168.2.1  
D地192.168.4.X  
1 D地局域网任何一台电脑ping192.168.0.1不通  
2 A地192.168.0.X平D地局域网内电脑有的通有的不通。  
3 2160内ping100E，192.168.0.X是通的。  
4 D地局域网电脑ping 192.168.1.1、192.168.2.1是通的。  
5 D地2160直接接电脑也无法ping通A地任何设备。  
华为工程师也看过两地配置，并且在他们的大实验室测试过是没问题的。并且说远程D地电脑时，有一次可以ping通A地服务器，但延迟较大。但是最近两天我试了很多次，一直无法ping通。  
顺便说一下，各地上外网都正常。  
小弟表达的不够清楚，大家有不明白的问我，我再一一回答，附上两地配置。  
**A地**  
Login authentication  
  
Username:  
Password:  
Note: The max number of VTY users is 5, and the current number  
      of VTY users on line is 1.  
<jinshengshuini-fr>sys  
Enter system view , return user view with Ctrl+Z.  
[jinshengshuini-fr]dis cur  
#  
sysname jinshengshuini-fr  
#  
web-manager enable  
web-manager security enable port 8089  
#  
l2tp enable  
#  
firewall packet-filter default permit interzone local trust direction inbound  
firewall packet-filter default permit interzone local trust direction outbound  
firewall packet-filter default permit interzone local untrust direction inbound  
firewall packet-filter default permit interzone local untrust direction outboun  
d  
firewall packet-filter default permit interzone local dmz direction inbound  
firewall packet-filter default permit interzone local dmz direction outbound  
firewall packet-filter default permit interzone trust untrust direction inbound  
firewall packet-filter default permit interzone trust untrust direction outboun  
d  
firewall packet-filter default permit interzone trust dmz direction inbound  
firewall packet-filter default permit interzone trust dmz direction outbound  
firewall packet-filter default permit interzone dmz untrust direction inbound  
firewall packet-filter default permit interzone dmz untrust direction outbound  
#  
nat server protocol tcp global xxx.xx.xx.a www inside 192.168.0.200 www  
nat server protocol tcp global xxx.xx.xx.a 3389 inside 192.168.0.200 3389  
nat server protocol tcp global xxx.xx.xx.a ftp inside 192.168.0.200 ftp  
nat server protocol tcp global xxx.xx.xx.a 5001 inside 192.168.0.200 5001  
nat server protocol tcp global xxx.xx.xx.a 81 inside 192.168.0.200 81  
nat server protocol tcp global xxx.xx.xx.a 82 inside 192.168.0.200 82  
nat server protocol tcp global xxx.xx.xx.a 83 inside 192.168.0.200 83  
nat server protocol tcp global xxx.xx.xx.a 84 inside 192.168.0.200 84  
nat server protocol tcp global xxx.xx.xx.a 85 inside 192.168.0.200 85  
nat server protocol tcp global xxx.xx.xx.a 86 inside 192.168.0.200 86  
nat server protocol tcp global xxx.xx.xx.a 87 inside 192.168.0.200 87  
nat server protocol tcp global xxx.xx.xx.a 88 inside 192.168.0.200 88  
nat server protocol tcp global xxx.xx.xx.a 89 inside 192.168.0.200 89  
nat server protocol tcp global xxx.xx.xx.a 90 inside 192.168.0.224 90  
nat server protocol tcp global xxx.xx.xx.a 91 inside 192.168.0.200 91  
nat server protocol tcp global xxx.xx.xx.a 92 inside 192.168.0.200 92  
nat server protocol tcp global xxx.xx.xx.a 93 inside 192.168.0.200 93  
nat server protocol tcp global xxx.xx.xx.a 94 inside 192.168.0.200 94  
nat server protocol tcp global xxx.xx.xx.a 95 inside 192.168.0.200 95  
nat server protocol tcp global xxx.xx.xx.a 96 inside 192.168.0.200 96  
nat server protocol tcp global xxx.xx.xx.a 97 inside 192.168.0.200 97  
nat server protocol tcp global xxx.xx.xx.a 98 inside 192.168.0.200 98  
nat server protocol tcp global xxx.xx.xx.a 99 inside 192.168.0.200 99  
nat server protocol tcp global xxx.xx.xx.a 100 inside 192.168.0.200 100  
nat server protocol tcp global xxx.xx.xx.a 3000 inside 192.168.0.200 3000  
nat server protocol tcp global xxx.xx.xx.a 3001 inside 192.168.0.200 3001  
nat server protocol tcp global xxx.xx.xx.a 3390 inside 192.168.0.224 3390  
nat server protocol tcp global xxx.xx.xx.a 4660 inside 192.168.0.224 4660  
nat server protocol tcp global xxx.xx.xx.a 4001 inside 192.168.0.224 4001  
undo nat alg enable esp  
nat alg enable ftp  
nat alg enable dns  
nat alg enable icmp  
nat alg enable netbios  
undo nat alg enable h323  
undo nat alg enable hwcc  
undo nat alg enable ils  
undo nat alg enable pptp  
undo nat alg enable qq  
undo nat alg enable msn  
undo nat alg enable user-define  
undo nat alg enable sip  
undo nat alg enable mgcp  
undo nat alg enable mms  
undo nat alg enable sqlnet  
undo nat alg enable rtsp  
firewall permit sub-ip  
firewall fragment-discard enable  
#  
firewall statistic system enable  
#  
interface Aux0  
async mode flow  
link-protocol ppp  
#  
interface Ethernet0/0/0  
description Connect to ISP  
ip address xxx.xx.xx.a 255.255.255.252  
#  
interface Ethernet0/0/1  
description Connect to localnetwork  
ip address 192.168.0.1 255.255.255.0  
undo ip fast-forwarding qff  
#  
interface Ethernet0/0/2  
#  
interface Ethernet0/0/3  
#  
interface Ethernet0/0/4  
#  
interface Virtual-Template1  
ppp authentication-mode chap  
ip address 192.168.3.1 255.255.255.0  
remote address pool 1  
#  
interface Tunnel0  
description Connect dongtai  
ip address 192.168.10.1 255.255.255.0  
source xxx.xx.xx.a  
destination xxx.xx.xx.b  
#  
interface Tunnel1  
description Connect to guyuan  
ip address 192.168.11.1 255.255.255.0  
source xxx.xx.xx.a  
destination xxx.xx.xx.c  
#  
interface Tunnel3  
description Connect to yinxin  
ip address 192.168.13.1 255.255.255.0  
source xxx.xx.xx.a  
destination xxx.xx.xx.d  
#  
interface Secp0/0/0  
#  
interface NULL0  
#  
right-manager server-group  
#  
acl number 2001  
rule 0 permit source 192.168.0.200 0  
rule 1 permit source 192.168.0.201 0  
rule 2 permit source 192.168.0.202 0  
rule 3 permit source 192.168.0.203 0  
rule 4 permit source 192.168.0.204 0  
rule 5 permit source 192.168.0.205 0  
rule 6 permit source 192.168.0.206 0  
rule 7 permit source 192.168.0.207 0  
rule 8 permit source 192.168.0.208 0  
rule 9 permit source 192.168.0.209 0  
rule 10 permit source 192.168.0.210 0  
rule 11 permit source 192.168.0.211 0  
rule 12 permit source 192.168.0.212 0  
rule 13 permit source 192.168.0.213 0  
rule 14 permit source 192.168.0.214 0  
rule 15 permit source 192.168.0.215 0  
rule 16 permit source 192.168.0.216 0  
rule 17 permit source 192.168.0.217 0  
rule 18 permit source 192.168.0.218 0  
rule 19 permit source 192.168.0.219 0  
rule 20 permit source 192.168.0.220 0  
rule 21 permit source 192.168.0.221 0  
rule 22 permit source 192.168.0.222 0  
rule 23 permit source 192.168.0.223 0  
rule 24 permit source 192.168.0.224 0  
rule 25 permit source 192.168.0.225 0  
rule 26 permit source 192.168.0.226 0  
rule 27 permit source 192.168.0.227 0  
rule 28 permit source 192.168.0.228 0  
rule 29 permit source 192.168.0.229 0  
rule 30 permit source 192.168.0.230 0  
rule 31 permit source 192.168.0.231 0  
rule 32 permit source 192.168.0.232 0  
rule 33 permit source 192.168.0.233 0  
rule 34 permit source 192.168.0.234 0  
rule 35 permit source 192.168.0.235 0  
rule 36 deny  
#  
acl number 3001  
rule 0 permit tcp source-port eq 95 destination xxx.xx.xx.a 0 destination-por  
t eq 95  
rule 1 permit tcp source-port eq 96 destination xxx.xx.xx.a 0 destination-por  
t eq 96  
rule 2 permit tcp source-port eq 97 destination xxx.xx.xx.a 0 destination-por  
t eq 97  
rule 3 permit tcp source-port eq 98 destination xxx.xx.xx.a 0 destination-por  
t eq 98  
rule 4 permit tcp source-port eq 99 destination xxx.xx.xx.a 0 destination-por  
t eq 99  
rule 5 permit tcp source-port eq 100 destination xxx.xx.xx.a 0 destination-po  
rt eq 100  
rule 6 permit tcp source-port eq ftp destination xxx.xx.xx.a 0 destination-po  
rt eq ftp  
rule 7 permit tcp source-port eq 5001 destination xxx.xx.xx.a 0 destination-p  
ort eq 5001  
rule 8 permit tcp source-port eq www destination xxx.xx.xx.a 0 destination-po  
rt eq www  
rule 9 permit tcp source-port eq 81 destination xxx.xx.xx.a 0 destination-por  
t eq 81  
rule 10 permit tcp source-port eq 82 destination xxx.xx.xx.a 0 destination-po  
rt eq 82  
rule 11 permit tcp source-port eq 83 destination xxx.xx.xx.a 0 destination-po  
rt eq 83  
rule 12 permit tcp source-port eq 84 destination xxx.xx.xx.a 0 destination-po  
rt eq 84  
rule 13 permit tcp source-port eq 85 destination xxx.xx.xx.a 0 destination-po  
rt eq 85  
rule 14 permit tcp source-port eq 86 destination xxx.xx.xx.a 0 destination-po  
rt eq 86  
rule 15 permit tcp source-port eq 87 destination xxx.xx.xx.a 0 destination-po  
rt eq 87  
rule 16 permit tcp source-port eq 88 destination xxx.xx.xx.a 0 destination-po  
rt eq 88  
rule 17 permit tcp source-port eq 89 destination xxx.xx.xx.a 0 destination-po  
rt eq 89  
rule 18 permit tcp source-port eq 90 destination xxx.xx.xx.a 0 destination-po  
rt eq 90  
rule 19 permit tcp source-port eq 3000 destination xxx.xx.xx.a 0 destination-  
port eq 3000  
rule 20 permit tcp source-port eq 92 destination xxx.xx.xx.a 0 destination-po  
rt eq 92  
rule 21 permit tcp source-port eq 93 destination xxx.xx.xx.a 0 destination-po  
rt eq 93  
rule 22 permit tcp source-port eq 94 destination xxx.xx.xx.a 0 destination-po  
rt eq 94  
rule 23 permit tcp source-port eq 8089 destination xxx.xx.xx.a 0 destination-  
port eq 8089  
rule 24 permit tcp source-port eq 91 destination xxx.xx.xx.a 0 destination-po  
rt eq 91  
rule 25 permit tcp source-port eq 3001 destination xxx.xx.xx.a 0 destination-  
port eq 3001  
acl number 3999  
rule 5 permit ip source 192.168.4.1 0 destination 192.168.0.1 0  
rule 10 permit ip source 192.168.0.1 0 destination 192.168.4.1 0  
#  
firewall zone local  
set priority 100  
#  
firewall zone trust  
set priority 85  
add interface Ethernet0/0/1  
#  
firewall zone untrust  
set priority 5  
add interface Ethernet0/0/0  
add interface Tunnel0  
add interface Tunnel1  
add interface Tunnel3  
#  
firewall zone dmz  
set priority 50  
#  
firewall interzone local trust  
#  
firewall interzone local untrust  
#  
firewall interzone local dmz  
#  
firewall interzone trust untrust  
packet-filter 3001 inbound  
nat outbound 2001 interface Ethernet0/0/0  
#  
firewall interzone trust dmz  
#  
firewall interzone dmz untrust  
#  
l2tp-group 1  
mandatory-lcp  
undo tunnel authentication  
allow l2tp virtual-template 1  
#  
aaa  
local-user   
local-user   
local-user   
local-user   
ip pool 1 192.168.3.2 192.168.3.254  
#  
authentication-scheme default  
#  
authorization-scheme default  
#  
accounting-scheme default  
#  
domain default  
#  
#  
slb  
#  
ip route-static 0.0.0.0 0.0.0.0 xxx.xx.xx.xx  
ip route-static 192.168.1.0 255.255.255.0 Tunnel0  
ip route-static 192.168.2.0 255.255.255.0 Tunnel1  
ip route-static 192.168.4.0 255.255.255.0 Tunnel3  
#  
user-interface con 0  
authentication-mode aaa  
lock lock-timeout 5  
set authentication password simple xxx  
idle-timeout 0 0  
user-interface aux 0  
user-interface vty 0  
authentication-mode aaa  
set authentication password simple axxxx  
user-interface vty 1 4  
#  
return  
[jinshengshuini-fr]  
  
  
  
  
  
  
**D地**  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*          All rights reserved (2008-2009)               \*  
\*      Without the owner's prior written consent,        \*  
\* no decompiling or reverse-engineering shall be allowed.\*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
NOTICE:This is a private communication system.  
       Unauthorized access or use may lead to prosecution.  
  
Login authentication  
  
Username:  
Password:  
Note: The max number of VTY users is 5, and the current number  
      of VTY users on line is 1.  
<YinXin>dis cur  
#  
sysname YinXin  
#  
web-manager enable  
#  
info-center source default channel 4 log level notifications  
#  
firewall packet-filter default permit interzone local trust direction inbound  
firewall packet-filter default permit interzone local trust direction outbound  
firewall packet-filter default permit interzone local untrust direction inbound  
firewall packet-filter default permit interzone local untrust direction outboun  
d  
firewall packet-filter default permit interzone local dmz direction inbound  
firewall packet-filter default permit interzone local dmz direction outbound  
firewall packet-filter default permit interzone trust untrust direction inbound  
firewall packet-filter default permit interzone trust untrust direction outboun  
d  
firewall packet-filter default permit interzone trust dmz direction inbound  
firewall packet-filter default permit interzone trust dmz direction outbound  
firewall packet-filter default permit interzone dmz untrust direction inbound  
firewall packet-filter default permit interzone dmz untrust direction outbound  
#  
nat server protocol tcp global xxx.xx.xx.xxx 3389 inside 192.168.4.213 3389  
firewall permit sub-ip  
#  
firewall mode route  
#  
firewall statistic system enable  
#  
interface Cellular5/0/0  
link-protocol ppp  
#  
vlan 1  
#  
interface Vlanif1  
ip address 192.168.4.1 255.255.255.0  
undo ip fast-forwarding qff  
#  
interface Ethernet0/0/0  
ip address xxx.xx.xx.xxx 255.255.255.252  
#  
interface Ethernet1/0/0  
#  
interface Ethernet1/0/1  
#  
interface Ethernet1/0/2  
#  
interface Ethernet1/0/3  
#  
interface Ethernet1/0/4  
#  
interface Ethernet1/0/5  
#  
interface Ethernet1/0/6  
#  
interface Ethernet1/0/7  
#  
interface Tunnel3  
description Connect to shengjing  
ip address 192.168.13.2 255.255.255.0  
source xxx.xx.xx.d  
destination xxx.xx.xx.a  
#  
interface Tunnel4  
description connect to guyuan  
ip address 192.168.14.1 255.255.255.0  
source xxx.xx.xx.d  
destination xxx.xx.xx.b  
#  
interface Tunnel5  
description connect to dongtai  
ip address 192.168.15.1 255.255.255.0  
source xxx.xx.xx.d  
destination 1xxx.xx.xx.c  
#  
interface NULL0  
#  
right-manager server-group  
#  
acl number 2000  
rule 5 permit  
#  
acl number 3000  
rule 5 permit ip source 192.168.4.208 0.0.0.15  
#  
cwmp  
#  
firewall zone local  
set priority 100  
#  
firewall zone trust  
set priority 85  
add interface Vlanif1  
#  
firewall zone untrust  
set priority 5  
add interface Ethernet0/0/0  
add interface Tunnel5  
add interface Tunnel3  
add interface Tunnel4  
#  
firewall zone dmz  
set priority 50  
#  
firewall interzone local trust  
#  
firewall interzone local untrust  
#  
firewall interzone local dmz  
#  
firewall interzone trust untrust  
packet-filter 2000 inbound  
packet-filter 2000 outbound  
nat outbound 3000 interface Ethernet0/0/0  
#  
firewall interzone trust dmz  
#  
firewall interzone dmz untrust  
#  
aaa  
local-user   
local-user   
local-user   
local-user   
local-user   
local-user   
local-user   
local-user   
local-user   
authentication-scheme default  
#  
authorization-scheme default  
#  
accounting-scheme default  
#  
domain default  
#  
#  
slb  
#  
ip route-static 0.0.0.0 0.0.0.0 xxx.xx.xx.xxx  
ip route-static 192.168.0.0 255.255.255.0 Tunnel3  
ip route-static 192.168.1.0 255.255.255.0 Tunnel5  
ip route-static 192.168.2.0 255.255.255.0 Tunnel4  
#  
user-interface con 0  
user-interface tty 81  
authentication-mode none  
modem both  
user-interface vty 0 4  
authentication-mode aaa  
#  
return  
<YinXin>  
  
**补充B地配置**   
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*  Copyright(c) 1998-2007 Huawei Technologies Co., Ltd.  All rights reserved.  \*  
\*  Without the owner's prior written consent,                                  \*  
\*  no decompiling or reverse-engineering shall be allowed.                     \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
Login authentication  
  
Username:  
Password:  
<Quidway>dis cur  
#  
sysname Quidway  
#  
cpu-usage cycle 1min  
#  
firewall enable  
#  
radius scheme system  
#  
domain system  
#  
local-user   
password   
service-type telnet terminal  
level 3  
service-type ftp  
local-user   
password   
service-type telnet  
level 3  
#  
acl number 2000  
rule 0 permit source 192.168.1.0 0.0.0.255  
acl number 2001  
rule 0 permit source 192.168.1.200 0  
rule 1 permit source 192.168.1.201 0  
rule 2 permit source 192.168.1.202 0  
rule 3 permit source 192.168.1.203 0  
rule 4 permit source 192.168.1.204 0  
rule 5 permit source 192.168.1.205 0  
rule 6 permit source 192.168.1.206 0  
rule 7 permit source 192.168.1.207 0  
rule 8 permit source 192.168.1.208 0  
rule 9 permit source 192.168.1.209 0  
rule 10 permit source 192.168.1.210 0  
rule 11 permit source 192.168.1.211 0  
rule 12 permit source 192.168.1.212 0  
rule 13 permit source 192.168.1.213 0  
rule 14 permit source 192.168.1.214 0  
rule 15 permit source 192.168.1.215 0  
rule 16 permit source 192.168.1.216 0  
rule 17 permit source 192.168.1.217 0  
rule 18 permit source 192.168.1.218 0  
rule 19 permit source 192.168.1.219 0  
rule 20 permit source 192.168.1.220 0  
rule 21 permit source 192.168.1.221 0  
rule 22 permit source 192.168.1.222 0  
rule 23 permit source 192.168.1.223 0  
rule 24 permit source 192.168.1.224 0  
rule 25 permit source 192.168.1.225 0  
rule 26 permit source 192.168.1.226 0  
rule 27 permit source 192.168.1.227 0  
rule 28 permit source 192.168.1.228 0  
rule 29 permit source 192.168.1.229 0  
rule 30 permit source 192.168.1.230 0  
rule 31 permit source 192.168.1.231 0  
rule 32 permit source 192.168.1.232 0  
rule 33 permit source 192.168.1.233 0  
rule 34 permit source 192.168.1.234 0  
rule 35 permit source 192.168.1.235 0  
rule 37 permit source 192.168.1.252 0  
rule 38 permit source 192.168.1.253 0  
rule 39 deny  
#  
interface Aux0  
async mode flow  
#  
interface Ethernet0/0  
description Connect to isp  
ip address xx.xx.xx.b 255.255.255.240  
nat outbound 2001  
nat server protocol tcp global xx.xx.xx.b 10001 inside 192.168.1.252 10001  
nat server protocol udp global xx.xx.xx.b 10001 inside 192.168.1.252 10001  
nat server protocol tcp global xx.xx.xx.b 10002 inside 192.168.1.253 10002  
nat server protocol udp global xx.xx.xx.b 10002 inside 192.168.1.253 10002  
#  
interface Ethernet0/1  
description Connect to localnetwork  
tcp mss 1024  
ip address 192.168.1.1 255.255.255.0  
#  
interface Serial0/0  
clock DTECLK1  
link-protocol ppp  
ip address dhcp-alloc  
#  
interface Tunnel0  
description Connect to shengjin  
tcp mss 1024  
ip address 192.168.10.2 255.255.255.0  
source xx.xx.xx.b  
destination xx.xx.xx.a  
#  
interface Tunnel2  
description Connect to guyuan  
ip address 192.168.12.2 255.255.255.0  
source xx.xx.xx.b  
destination xx.xx.xx.c  
#  
interface Tunnel5  
description Connect to yinxin  
tcp mss 1024  
ip address 192.168.15.2 255.255.255.0  
source xx.xx.xx.b  
destination xx.xx.xx.d  
#  
interface NULL0  
#  
FTP server enable  
#  
ip route-static 0.0.0.0 0.0.0.0 xx.xx.xx.xx preference 60  
ip route-static 192.168.0.0 255.255.255.0 Tunnel 0 preference 60  
ip route-static 192.168.2.0 255.255.255.0 Tunnel 2 preference 60  
ip route-static 192.168.4.0 255.255.255.0 Tunnel 5 preference 60  
#  
user-interface con 0  
user-interface aux 0  
user-interface vty 0  
authentication-mode scheme  
set authentication password   
user-interface vty 1 4  
authentication-mode scheme  
#  
return  
<Quidway>

# **案例分析3**

USG6680 通过调整TCP MSS 规避大包不通的问题

|  |  |
| --- | --- |
| |  | | --- | | 问题描述  http://support.huawei.com/enterprise/product/images/55eb78b2e77c47f49484528d815123d8     俩台NE40E和俩台USG6680口字型组网，NE40E和USG设备之间经过运行商传输设备，俩台USG下带多台服务器，网络正常情况下业务上行主走USG6680-1的1/0/3，将1/0/3接口down之后，业务应绕行USG6680-2。当主链路domn之后，业务不通。  处理过程  1、 查看USG6680-2接口的MTU值为1500，ping测试USG-2和NE40E-2的互联IP测试，发现超过1470字节以上的大包不通，怀疑是传输侧对大包处理有问题，会丢弃超过1470字节以上的大包。  2、 因为USG下带的业务流量主要是TCP承载，通过在俩台USG6680设备上配置firewall tcp-mss 1300命令，保证经过USG6680转发的TCP报文数据长度为1300，加上IP头和TCP头的长度不超过1470字节，可以正常转发，规避了问题。  根因  传输侧对大包处理有问题，会丢弃超过1470字节以上的大包。  解决方案   通过配置firewall tcp-mss 1300之后，业务恢复正常。ping大包依然不通，由于ICMP报文不涉及业务不影响现网使用。  建议与总结      通过配置firewall tcp-mss 1300命令保证TCP报文不超过传输侧限制只能规避问题，建议客户和运行商协调解决传输侧大包不通的问题。 | |

# **案例分析4**

   在给某市工商局的一个项目中，实现分局的内外网分离，外网用的是H3C MSR20-20路由器作为网关，通过GRE隧道连接到市局。  
      在部署第一个分局时，就遇到了一个奇怪的问题，tunnel建立成功，ping网关、ping外网都是通的，但是就是网页打不开。。当时考虑是不是市局防火墙做了什么限制，把端口封掉了，于是在分居更换原来的2811，相同的配置，结果网页正常打开，那么问题肯定是出在这个MSR20-20上，在纠结很久无果后，打400电话，得知需要在接口下执行命令mss tcp xxx。。大概了解后，把mss值改为1300，结果正常访问。

pc访问http，大包情况下：data（1460）+tcp（20）+ip（20），到达msr后，加上gre（24印象中好像是28），由于接口mtu为1500，所以将数据包（1460+20+20+24=1524）分成2个包，分别作NAT（NAPT）到web站点，但tcp头部信息仅包含在第一个包中，2个包都达到目的地不能进行正常重组。如果在内网接口或者在tunnel接口在tcp mss调小，这个问题就ok了，其实还可以讲中间传输设备的mtu调大。至于tcp mss为什么是1300，这个应该是习惯问题，很多地方设置成1024或1400都是可以的，按照理论，此值最大应该可以设置成1500-20（ip）-24（gre）=1456。LZ可以去试试。。。

GRE封装后，会使原报文多24个字节，导致最终长度超过了MTU值，数据无法发送。将TCP mss改小后，报文最终长度小于MTU，可以正常发送，所以不会有问题

# **案例分析5**

一、MTU的定义及sv n相关概念：

Mtu即最大传输单元，全称为Maximum Transmission Unit，是指通信协议的某一层上面所能通过的最大数据包大小（以字节为单位）。由于定义的模糊性，在此也介绍几个相关的名词，MRU、PMTU、[MSS](http://www.c114.net/keyword/MSS)和JUMBO FRAME，供大家甄别。

MRU即最大接收单元，全称为Maximum Receive Unit,与MTU相对，称为最大接收单元，目前也没有权威的标准定义，但许多文章中有这个名词。一台主机或[路由器](http://www.c114.net/keyword/%C2%B7%D3%C9%C6%F7)的MTU与MRU可以不一致。

PMTU，全称为 path maximum transmission unit,即路径MTU，把一条[IP](http://www.c114.net/keyword/IP)路径上MTU的最小值称为PMTU，PMTU是个理想化的概念，但目前业界没有有效的手段来实现PMTU的发现和更新。`

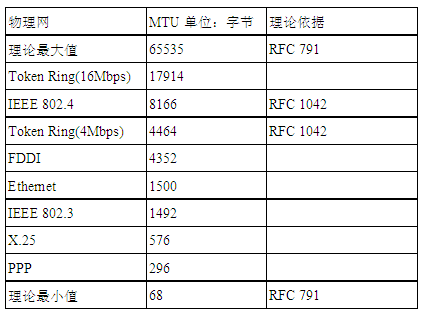
MSS是OSI参考模型中四层的一个概念，即最大分段长度，全称为TCP Maximum Segment Size，指TCP每次能够传输的最大数据分段长度（以字节为单位），MSS一般比MTU小40字节。

Jumbo Frame(有些称Giant Frame)，[网络](http://www.c114.net/keyword/%CD%F8%C2%E7)上会遇到jumbo frame的概念，[cisco](http://www.c114.net/keyword/cisco)路由器的接口中也有这个参数，超过[以太网](http://www.c114.net/keyword/%D2%D4%CC%AB%CD%F8)标准长度1518字节的帧称为jumbo frame。

二、MTU涉及主要原理：

1、常见网络的MTU值：

IP网络以包为单位进行信息传递，那么，一次传送多大的包合适、多大的包最高效就成为一个核心问题之一。MTU就是决定在什么样的物理网络传送多大数据包大的事实标准，不同类型网络由于物理特性、发展阶段不同，其MTU的默认值也不尽相同，以下是摘录的各类网络及其默认MTU值：



对于windows操作系统来讲，其以太网[网卡](http://www.c114.net/keyword/%CD%F8%BF%A8)MTU默认为1500，但可以通过修改工具或修改注册表进行修改，但只能改小，不能改大，即只能修改为小于或等于1500字节。

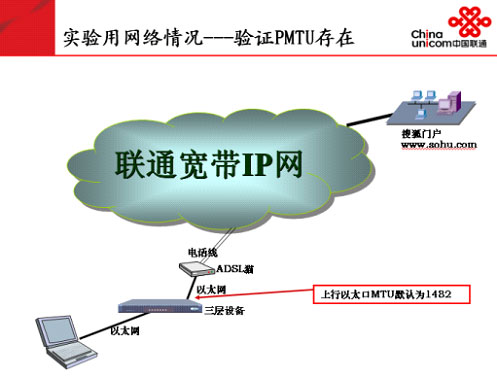
2、PMTU 发现过程：

对于一个基于网络的应用来讲，如果应用穿过网络的MTU与PMTU相等，那么应用穿过网络的效率最高，或者说，应用通过主机网卡发出的最大数据包与PMTU越接近(指小于等于PMTU)，应用穿过网络的效率越高，原因是有效的避免了分片和[重组](http://www.c114.net/keyword/%D6%D8%D7%E9)。

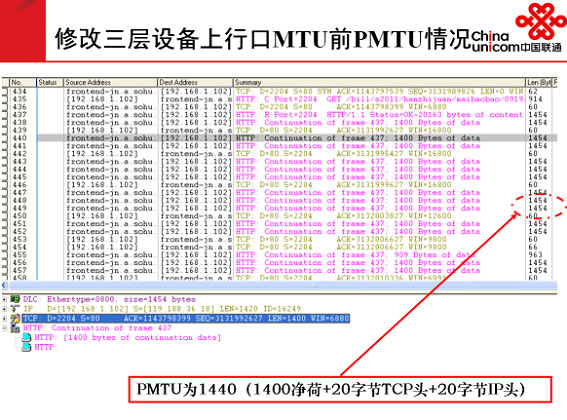
为了达到这个目的，一些操作系统支持自动发现路径MTU的功能，具体过程为：

路由器接口上收到一个报文长度大于本接口MTU值的报文，如果该报文被打上不分片的标记，将丢弃本报文，并且返回一个ICMP差错报文，通知报文发起者丢弃原因。报文发起者将发送比较小的报文。通过多次上述报文协商，将得到对于某一个固定路径上的最小Mtu值，这个过程叫做“Mtu Discovery”[详见RFC1191]。

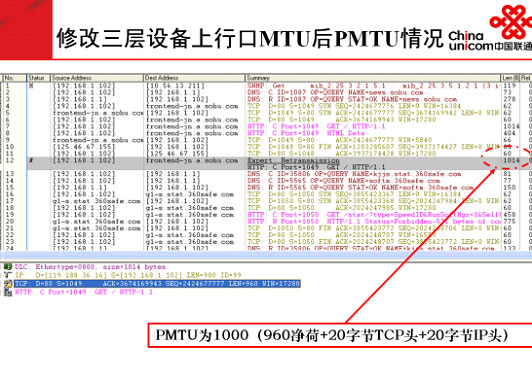
了解了MTU发现的原理，举一个实例验证PMTU变化过程：



在上图所示实验网络中，由三层设备模拟PPPOE拨号，实现接入[宽带](http://www.c114.net/keyword/%BF%ED%B4%F8)IP网。三层设备上行以太网口默认MTU为1482字节。抓包结果显示如下：



将三层设备上行以太网口默认MTU改为1000字节。抓包结果显示如下：



3、“PMTU”发现存在的问题：

由于[互联网](http://www.c114.net/keyword/%BB%A5%C1%AA%CD%F8)上路由器或其它网络设备的配置的无法统一规范，某些[运营商](http://www.c114.net/keyword/%D4%CB%D3%AA%C9%CC)或网站考虑到[网络安全](http://www.c114.net/keyword/%CD%F8%C2%E7%B0%B2%C8%AB)和其它需要，有时会把ICMP报文过滤掉，此外，PMTU牵涉到主机、各类[交换机](http://www.c114.net/keyword/%BD%BB%BB%BB%BB%FA)、路由器、防火墙等网络设备，这些主机和网络设备没有有效的手段实现PMTU的协商和交互,这样Mtu Discovery不能正常运行，影响应用正常运行，即实质上目前没有有效的手段来发现PMTU。

互联网上的网络设备,遇到MTU发现报文或必须将IP包分片但DF设置为1时，路由器可采用以下任一种方式(从网上摘录)：

发送符合 RFC 792 中最初定义的“ICMP Destination Unreachable-Fragmentation Needed and DF Set”消息，然后丢弃该包。原始消息格式中不包含有关转发失败的链路的 IP MTU 的信息。（导致PMTU无法正常发现）  
⎫ 发送符合 RFC 1191 中重新定义的“ICMP Destination Unreachable-Fragmentation Needed and DF Set”消息，然后丢弃该包。此新消息格式包含一个 MTU 字段，可指出转发失败的链路的 IP MTU。（PMTU可能会正常发现）

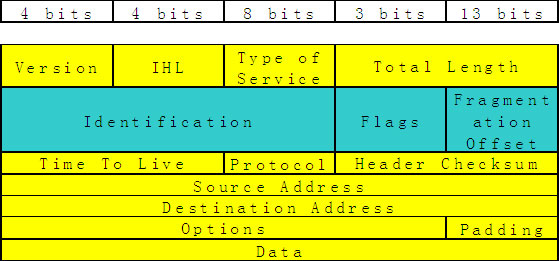
RFC 1191 定义了路径 MTU (PMTU) 发现，它使得源和目的 TCP 对等方能够动态地发现二者之间路径的IP MTU，从而发现该路径的 TCP MSS。一旦收到符合 RFC 1191 定义的“Destination Unreachable-Fragmentation Needed and DF Set”消息，TCP 就会将该连接的 MSS 调整为指定 IP MTU 减去 TCP 和 IP 报头的大小。这样，在该 TCP 连接上发送的后续包就不会超过最大大小，无需分段即可在该路径上传输。

直接丢弃包。直接丢弃需分段但 DF 标记设置为 1 的包的路由器称为 PMTU 黑洞路由器。

总之，PMTU的不可发现性，导致因MTU问题引起的应用系统无法正常运行情况时有发生。

4、超过MTU值的数据包分片、重组过程，：

IP包的格式如下：



IP包的分片重组牵涉到IP包头的几个重要字段，主要是标识符、标识位、偏移量，分别详述如下：

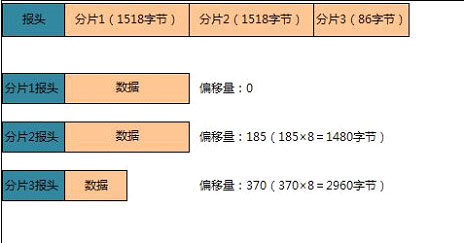
标识符（Identification）：在发送数据包前，发送主机给每个数据包一个ID值，放在16位的标识符字段中。此ID用于标识唯一的数据报或数据流。接收主机利用此ID对收到的数据报进行重组。当分片的IP数据报从源地址发送到目的地址的时候，由于网络延迟或者不同的传输路径的关系，在到达目的主机时，这些分片数据报并不总是有序的到达，而是处于一种无序状态，因此，接收主机便用此ID判断接收的这些分片数据报是否属于同一个数据流，然后再进行重组。

标志（Flags）：第一个bit称为R位，目前保留未用。第二个bit称为DF位，Don't Fragment，“不分片”位，即如果将这一比特置1 ，表示上层应用不允许分片，IP层将不对数据报进行分片。第三个bit称为MF位，More Fragment，“更多的片”。除了最后一个数据分片外，其他每个组成数据报的数据分片都要把该比特置1，表示有更多的分片。

偏移量(Framentation offset)：13位的偏移量字段用来表示分段的数据报在整个数据流中的位置，即相当于分片数据报的顺序号。发送主机对第一个数据报的偏移量置为0，而后续的分片数据报的偏移量则以网络的MTU大小赋值。偏移量对于接收方进行数据重组的时候，这是一个唯一依据。对于分片的数据段(单位：字节)必须为8的整数倍，否则IP无法表达其偏移量。

了解了数据分片的几个关键字段后，我们找一个实例说明分片过程（从网上摘录）：

在MTU为1500的以太网中，源主机如果需要通过UDP传送3000字节的数据到目的主机，这时的分段情况如下所示(假定在同一网段)：



此处需要注意的是对于分片1的报头，相对于其他两个分片的报头要多8个字节UDP协议报头的开销。因此，在计算实际传输的数据净载荷时，分片1要多减去8字节UDP报头。最后，接收主机通过包ID、标识位、偏移量值将数据重组成完整的数据。

需要注意是，有些数据包在一次分片后，由于遇到更小MTU的网络，还可能被继续分片。为数据包分片和为数据包再次分片的区别在于[网关](http://www.c114.net/keyword/%CD%F8%B9%D8)处理MF位的不同。在一个网关为原来未分片的数据包分片时，除了末尾的数据包片，它将其余所有分片上的MF位置为1，最后一片为0。然而，当网关为一个非末尾的数据包片再次分片时，它会把生成的所有子分片中的MF位全部调为1，因为所有这些子分片都不可能是整个数据包末尾的数据包。分片中标志字段的M值取决于该分片是否是原始分组的最后一片，而片偏移量也是相对于原始分组的偏移量。

三、MTU对上层应用的影响：

1、TCP、UDP等上层应用传输效率的高低与PMTU有密切联系。

对TCP来讲，其传输效率与MSS的合适大小密切相关，而MTU是决定MSS大小的唯一因素。MSS的大小会在TCP连接建立阶段进行协商，具体协商过程如下：

TCP client发出SYN报文，其中option选项填充的MSS字段一般为(MTU-IP头大小-TCP头大小)，同样TCP server收到SYN报文后，会发送SYN＋ACK报文应答，option选项填充的mss字段也为(MTU-IP头大小-TCP头大小)；协商双方会比较SYN和SYN+ACK报文中MSS字段大小，选择较小的MSS作为发送TCP分片的大小。

理论上，如果MSS与合适的MTU即PMTU（我认为合适的PMTU为路径上可不分片通过的最大MTU）匹配，TCP传输效率最高，因为免去了分片、重组等工作。但由于PMTU发现过程无法保证，导致最终发现的MTU可能并不合适路径传输，造成部分应用的分片和重组，降低了传输效率。

2、部分网络设备对特定应用的分片报文处理能力弱，造成部分应用故障。

目前，国内宽带IP网络多数用户采用PPPOE拨号方式实现业务接入。位于[电信](http://www.c114.net/keyword/%B5%E7%D0%C5)运营商内侧的BAS设备用于终结PPP连接，实现用户接入汇聚。实际应用中，部分BAS设备对[VPDN](http://www.c114.net/keyword/VPDN)等特定应用的分片、重组机制不健全，造成部分基于VPDN方式的业务速度缓慢。如：笔者曾发现过某厂商BAS设备ME60当TSU板未启用时，其于L2TP隧道的VPDN拨号应用，经常出现业务访问缓慢，甚至不通的情况，当启用TSU板后，业务恢复正常（经了解，TSU板是专门用于基于隧道或其它协议数据包分片处理功能）。某厂商设备SE800，承载了基于L2TP隧道的VPDN拨号应用后，部分应用可以使用，部分应用无法正常运行。

3、部分网络安全设备对分片的重组过程进一步影响业务应用。

原则上，当某个网络应用业务在传输过程中发生分片后，只有发送端和接收端会进行分片的重组，这已经影响到业务运行速度。当分过片的IP包经过网络[监测](http://www.c114.net/keyword/%BC%E0%B2%E2)设备、安全系统等设备时，基于安全和其它特定目的，有些设备要求完成数据包的分片和重组，会进一步延缓应用的运行速度，甚至导致部分应用无法使用。

四、MTU常见问题小结(摘录自网络)：

1、为何有些共享上网的路由器的网络设备以太网口MTU值不是1500？

有些通过共享路由器PPPOE拨号上网的路由器上连口（与[ADSL](http://www.c114.net/keyword/ADSL)调制器互联的以太网口）MTU为1492，因为PPP报文占据了8个字节，导致承载数据信息的IP报文大小变成了1492。

2、为什么在[思科](http://www.c114.net/keyword/%CB%BC%BF%C6)路由器上GRE接口的默认MTU为1476？

因为GRE会重新封装一个IP包头，以及加上GRE的4字节头部，一共是24个字节。这样总的用于应用的IP包的长度为1476。

3、为什么支持[VLAN](http://www.c114.net/keyword/VLAN)标记的接口MTU要大于1514字节？

由于VLAN的原理是在以太网的帧头部加入了4个字节的VLAN TAG信息，因此在支持802.1q标记的链路的接口上要求MTU不小于1518字节，才能保证净荷为1500的数据包顺利通过该接口。

4、MTU和各种[VPN](http://www.c114.net/keyword/VPN)的关系有无规律性？

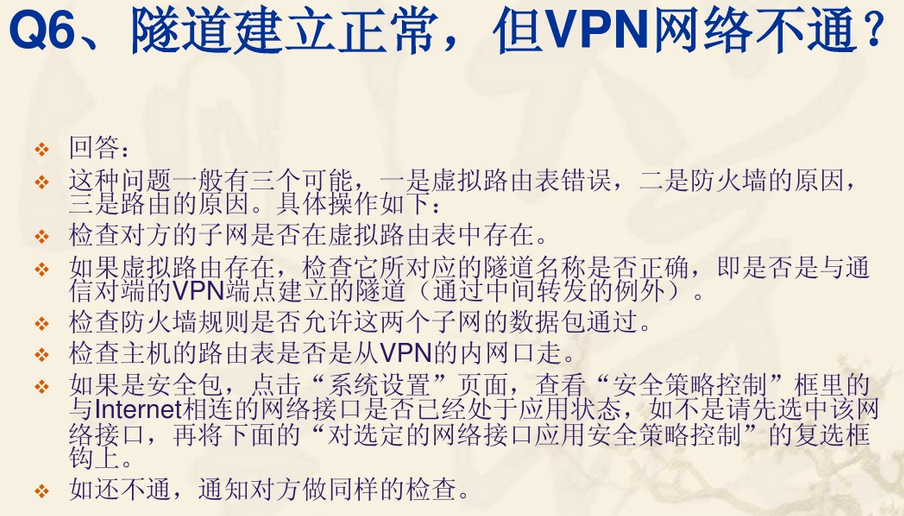
因VPN的实现需要对原有的IP或者是TCP/UDP数据进行封装，因此也就增加了数据包或数据帧的总长度，这样也就导致了VPN 承载的数据净荷值的减小，具体减少的数值与不同的VPN类型相关。

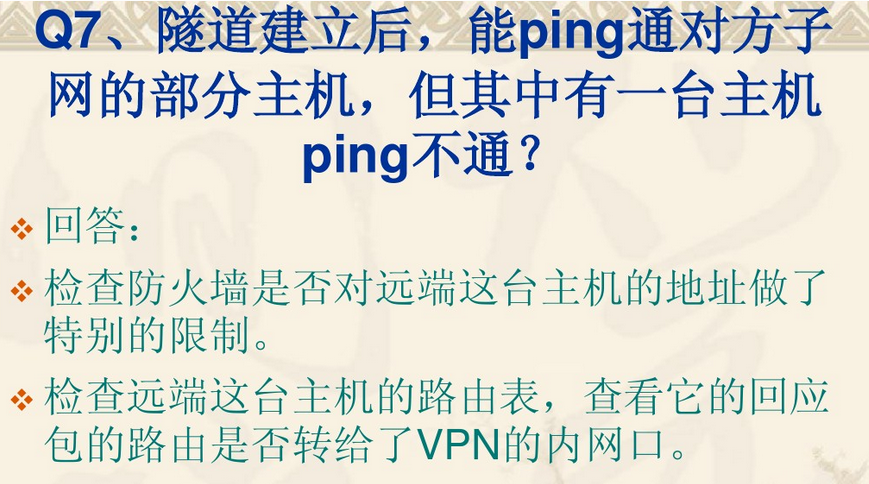
即：MTU=经VPN封装后数据包长度-VPN封装包头长度

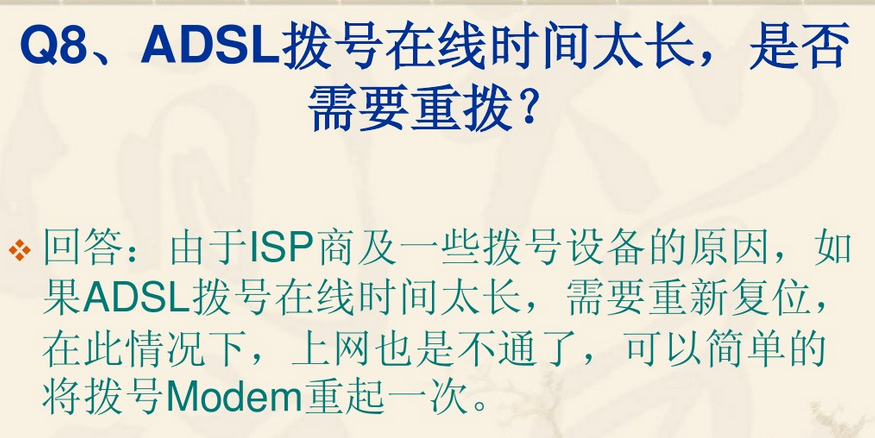
5、对于因MTU问题引起的应用系统故障现象有哪些？处理思路是什么？

因MTU引的网络问题表现各种各样，如：有些游戏，经常卡机。有的网站，部分网页可以打开，部分网页打不开。总的来讲，如果网络不丢包，而网络应用时快时慢，或部分应用无法使用，很大程度上可以定位为MTU问题。

对于MTU引起的问题，处理思路为：要么增加网络中最细管道的MTU值，要么减少终端应用发现的数据包大小。对于终端的MTU，会有各种各样的工具或方法进行修改







# **案例分析6**

负载均衡故障诊断：一个MSS值引发的疑案

1，此案例为服务器端系统异常MSS通告导致的业务应用故障，的确属于疑难故障范畴；

2，关于MSS的知识点，大家可参考本博客《[TCP MSS与PMTUD](http://www.vants.org/?post=109)》一文；

3，此案例充分证明，在遇到疑难故障时，站在原始报文交互的角度做深入的解码分析才是分析定位的根本之道；

4，此案例中，使用了正常时与异常时的对比分析法，关于对比分析法，大家可参考本博客《[疑难网络故障的分析方法和原理之对比分析法](http://www.vants.org/?post=49)》一文；

5，个人认为，此案例还可以做进一步的分析，如原作者开篇所言“很多东西都在互相影响”，既然经过负载均衡设备访问就出现这种问题，那么是否是负载均衡在发给服务器端的SYN报文跟客户端直接发给服务器端的SYN报文某些地方存在差异导致的呢？

**【原文链接】：**

<http://virtualadc.blog.51cto.com/3027116/692407>

**【原文全文】：**

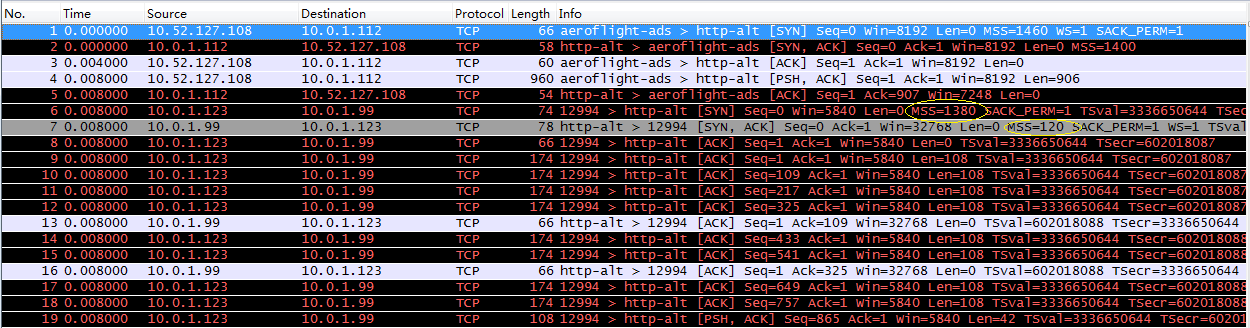
负载均衡设备位于客户端和真实服务器之间，一旦访问发生问题，在客户经过简单诊断后，负载均衡设备往往会成为首要被怀疑的对象。客户一般这样质疑：为什么我直接访问服务器没有问题，通过你的设备访问就不行了呢？ 质疑的确实有道理，但大多数事情往往不是非一即二这样简单，有很多东西都在互相影响，这就使得真相迷雾重重。

某一天接到某客户报障，说是通过负载均衡设备访问某一业务的时候，页面无法打开或者等半天后只打开了部分页面，而客户端如果直接访问服务器，则可以顺利打开页面。

事情很明显，这中间肯定是有问题存在。登录负载均衡设备检查配置和log，并取一些内部诊断信息，没发现什么错误，只剩下唯一的办法：去客户现场抓包分析。

于是开始抓包，同时抓回了出现问题的服务的数据包和其他没有出现问题的服务的数据包。

经过分析，果然有所不同，下面是有问题的抓包内容（抓包1）：

[](http://img1.51cto.com/attachment/201110/231444743.png)

10.52.127.108为客户端地址

10.0.1.112为VIP

10.0.1.99为真实服务器地址

由于是以旁路方式部署，需要转换源IP, 10.0.1.123为经过负载均衡设备转换的客户端地址（snat地址）

负载均衡的VIP配置为HTTP模式，这表示负载均衡设备是以proxy的方式来处理连接，也就是对每个连接，客户端先跟负载均衡设备完成一个三次握手，然后负载均衡设备再跟真实服务器完成一个三次握手。

访问流程：

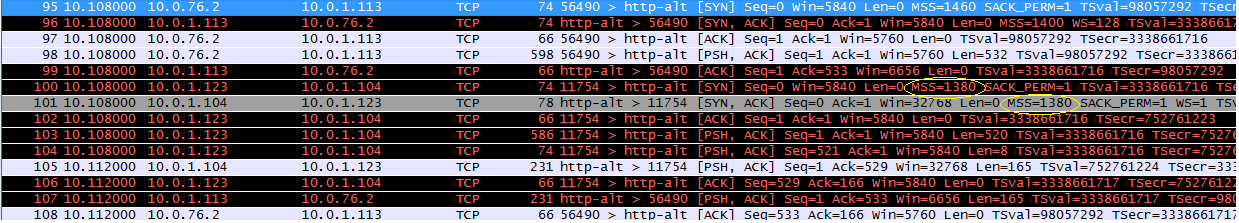
1) 10.52.127.108访问10.0.1.112

2) 负载均衡设备与客户端完成三次握手

3）然后负载均衡设备把源IP: 10.52.127.108转换成10.0.1.123向服务器10.0.1.99发起连接

4）服务器10.0.1.99与负载均衡设备完成三次握手。

下图是访问没有问题的服务的抓包内容（抓包2）：

[](http://img1.51cto.com/attachment/201110/231515640.png)

10.0.76.2为客户端地址

10.0.1.113为VIP

10.0.1.104为真实服务器地址

由于是以旁路方式部署，同样需要把客户端源IP转换为 10.0.1.123

访问流程跟抓包1相同。

仔细比较两个抓包内容，终于发现了差异出现在MSS值的协商上。

首先我们描述一下Client访问Server过程中MSS值的协行过程：

1） 客户端在向服务器发出SYN包的时候，会带上客户端设备可以接受的最大MSS值，意思是服务器发送到客户端的每个包的内容大小都不能大于这个值。

2） 服务器向客户端回复SYN,ACK包的时候，会比较客户端发来的MSS值和自己设定的MSS值，取两者的最小值作为自己可以接受的最大MSS值返回给客户端，意思是告诉客户端发送到服务器的每个包的内容大小都不能大于这个值。

3） 在实际的传输中，双方往往会取二者中的最小值作为双方互相发送的包大小的最大值。

基于以上通信流程我们来分析一下以上的两个抓包内容：

抓包1：

客户端发出SYN包，标明自己可接受的最大MSS值为1460,负载均衡设备回应自己可接受的MSS值为1400,协商成功后，双方交互的包大小不会大于1400。

负载均衡设备向服务器发出自己的可接受MSS值为1380,服务器回应自己可接受的MSS值120,协商成功后，负载均衡设备发给服务器的包就不能大于120了。

问题正是出在最后跟服务器协商出的大小为120的MSS值上。

我们看到客户端向负载均衡设备发出的第一个请求包大小为905字节，这个包大小不大于1400，所以负载均衡设备接收到了，接着负载均衡设备要把该请求发给选定的服务器10.0.1.99，由于服务器可接收的包不能大于120,所以负载均衡设备只能把客户端发来的请求包分成八个小包发送给服务器，然后一些不可控制的问题就出现了，客户端发出请求包后，需要等待应答，但由于负载均衡设备把一个包分解成8个包后，使得负载均衡设备跟服务器之间的交互时间变长，这个过程中客户端可能会超时重发请求包，而负载均衡设备跟服务器之间那八个小包的处理还可能出现丢包，重传，重装等问题。最关键是客户端在该连接的所有请求发完后如果是发送一个RST包来关闭连接，那么即使该连接上还有内容没传输完，该条连接也会关闭，由于一个请求包分成太多的小包传输，一旦发生客户端发出RST包的这种情况，基本上都会导致数据不能传输完毕，以上种种原因导致了页面不能打开或者不能完全打开的现象。

我们再分析抓包2：

客户端发出SYN包，标明自己可接受的最大MSS值为1460,负载均衡设备回应自己可接受的MSS值为1400,协商成功后，双方交互的包大小不会大于1400。这一点跟抓包1相同。

负载均衡设备向服务器发出自己的可接受最大MSS值为1380,服务器回应自己可接受的MSS值1380,协商成功后，所以双方会以1380的MSS值互相通信。

无论是客户端跟负载均衡设备还是负载均衡设备跟服务器之间，都是一个请求一个应答就能完成交互，不会发生要把包分割的现象，所以不会出现抓包1所出现的问题。

网络通信中由于MTU的设置不当引发的问题屡见不鲜，比如在存在ADSL设备的情况下，如果把设备的MTU设置成1500, 往往客户端的访问会出现问题，这是因为ADSL的PPPoE协议在MTU中占去8个字节，也就是ADSL的MTU最大值最多为1492, 如果客户端跟服务器设的很大，传输的数据包恰好大于1492字节，将导致数据包不能通过。 在程序设计中，程序所取MSS值往往是本机的MTU-40（TCP和IP头各占20个字节，MTU一般设成1500）， 所以基本上所有设备所能接受的最大MSS值不可能会大于1500-40=1460, 那么再考虑到网络中可能会存在PPPoE，VPN等设备会占用更多MTU字节，所以各家网络设备厂商提供的网络设备会进一步减小MSS值的设置，一般网络设备设定的MSS值大小为1400左右。

显然1400字节左右的MSS值是网络通信中的正常值，所以服务器返回一个120字节的MSS值这是一个不正常的现象，所以问题的根源在于服务器返回的MSS值不合适，那么这个值是谁返回的呢？ 是服务器，也就是说该返回哪个值主动权在于服务器，所以我们诊断问题原因出在服务器上。

接下来的处理需要去检查服务器为什么返回这个值，跟负载均衡设备无关了。但仍然有追踪的价值，因为服务器并不是一直返回120这个值，而是有些时候会协商成1380,这时候访问是正常的，有些时候是返回120，这时候就自然访问不正常。

客户的服务器装的是HP操作系统，应用软件是Oracle的ebs，在我们把问题定位到了服务器后，客户也找了HP的工程师来检查和分析，但无法找出原因。

个人分析问题原因可能出现在如下几个方面：

1） HP操作系统或者网卡驱动程序关于MTU的定义存在可变值，或者

2） Oracle ebs的底层通信程序在MSS值的协商时，会根据一些条件改变MSS值

以上仅仅是猜测，因为没有以上两个厂家的资深工程师的深度参与，无法最终定位结果，所以该问题成为了一个疑案。

# **案例分析7**

本文档将介绍 L2TP 链路中的分段与重组，并解释最大传输单元 (MTU)

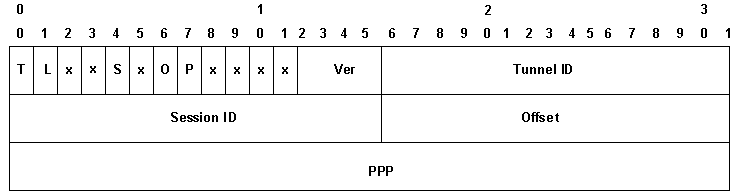
调整如何帮助减少一些相关问题。

## 分段示例

有时您需要对隧道封装的数据包进行分段以便在线路中进行传输。下面是一个相关示例。

在使用 L2TP over UDP 的情况下，所有协议的开销都包含一组额外的 IP、UDP 和 L2TP 报头。IP 报头是 20 字节，UDP 报头是 8 字节，L2TP 报头通常是 12 字节。12 字节的 L2TP 报头包含：

* 版本和标志字段（2 字节）
* 隧道 ID 和会话 ID 字段（各 2 字节）
* 2 字节的填充偏移量
* 4 字节的点对点协议 (PPP) 封



如果您启用数据序列化（默认情况下在 Cisco 设备中被禁用），您必须为 Ns 和 Nr 字段另外添加 4 字节。将 IP、UDP 和 L2TP 报头相加，可发现 L2TP over UDP 向数据包中添加了 40 字节的协议封装。

将 1500 字节的 IP 数据包封装到 L2TP 中时，封装的数据包变为 1540 字节（1500 + 40 字节的 IP、UDP 和 L2TP 报头）。您必须对数据包分段以便通过标准以太网类型接口（包含 1500 字节的 MTU）进行传输。封装的数据包分为两部分。第一个分段包含 1500 字节（1460 字节的原始 IP 数据包 + 40 字节的 L2TP 封装）。第二个分段包含 60 字节（最后 40 字节的原始 IP 数据包 + 20 字节的 IP 开销）。

**注意：**仅第一个分段包含 L2TP 报头；第二个分段仅包含 IP 报头。这允许 L2TP 对等体（LAC 或 LNS）将两个分段重组为 1540 字节的隧道封装的原始数据包。

**问题**

基于 User Datagram Protocol (UDP) 的第 2 层隧道协议 (L2TP) 以及其他第 2 层和第 3 层基于 IP 的隧道协议面临的问题之一是：隧道协议的开销增加了隧道封装的数据包大小。如果原始数据包已是最大尺寸，您必须将隧道封装的数据包分段以便在线路中进行传输。

分段与重组 L2TP 接入集中器 (LAC) 和 L2TP 网络服务器 (LNS) 中的 L2TP 数据包引起的问题之一是：分段与重组是在 Cisco IOS 软件中的进程级别完成的。当在 LNS 中聚集大量的 L2TP 会话和数据流时，进程交换可能会大大降低性能。出于该原因，减少或排除在 L2TP 交换路径中分段和重组的需求是非常必要的。

使用本文档中所述的方法之一调整最大传输单元 (MTU) 以解决该问题。

**MTU 调整方法**

Cisco IOS 软件中的多个配置和功能旨在通过调整 MTU 来避免在 L2TP 交换路径中分段和重组。

### 手工配置更小的 IP MTU

使用 **ip mtu 命令**在虚拟模板接口上配置较低的 IP MTU。配置较低的 IP MTU 以迫使路由器丢弃任何超过 IP MTU 并且已在 IP 报头中设置 DF（不分段）位的 IP 数据包。然后，路由器生成面向数据包源（原始主机）的“Internet Control Message Protocol (ICMP) type 3 Host Unreachable, code 4 fragmentation needed”消息。该消息指示接口的 IP MTU，以便源可以缩小数据包大小来适合接口。该进程还称为路径 MTU 检测 (PMTUD)。有关详细信息，请参阅 [RFC 1191](http://www.ietf.org/rfc/rfc1191.txt) 。[leavingcisco.com](http://www.cisco.com/swa/i/icon_popup_short.gif) [添加完整的 L2TP 报头时，将 IP MTU 配置为不超过 LAC 与 LNS 之间的 PMTU 的最大 IP 数据包大小。对于 1500 字节的 PMTU 和 40 字节的标准 L2TP 报头，请将 IP MTU 设置为 1460（1500-40 字节的报头）。](http://www.ietf.org/rfc/rfc1191.txt)

如果 LAC 与 LNS 之间的 PMTU 未知（或发生更改），您可以在 **vpdn-group** 下配置命令 ip pmtu 。Cisco IOS 软件版本 12.2(4)T 中已使用 Bug ID CSCds72714（外部用户不能查看）添加了 **ip pmtu 命令**。**ip pmtu** 功能将 DF 位从内部数据包复制到外部 L2TP 报头，并在路由器与其 L2TP 隧道终点之间启用 PMTUD。

### 调整 Windows PC 上的 PMTU

Microsoft Windows 包含允许您为 PMTU 发现启用补偿功能的注册表设置。有关 Windows NT 的信息，请参阅 Microsoft 网站上的下列文章：[Windows NT 3.51 的 PMTU 黑洞检测算法更改 (Q136970)](http://support.microsoft.com/default.aspx?scid=kb;en-us;Q136970)。[leavingcisco.com](http://www.cisco.com/swa/i/icon_popup_short.gif)

对于 Windows 2000/XP，Microsoft 文章[如何解决黑洞路由器问题 (Q314825)](http://support.microsoft.com/default.aspx?scid=kb;en-us;Q314825) 介绍了 Windows 中多种用于避免该问题的方法。[leavingcisco.com](http://www.cisco.com/swa/i/icon_popup_short.gif) [该文章对“黑洞”路由器进行了定义，介绍了定位黑洞路由器的方法，并给出了三种避免黑洞路由器可能会导致的数据丢失的建议方法。](http://support.microsoft.com/default.aspx?scid=kb;en-us;Q314825)

### 自动调整 IP MTU

您还可以启用 IP MTU 的自动调整。该功能允许路由器自动调整虚拟访问接口上的 IP MTU 以补充 L2TP 报头和输出接口的 MTU 的大小。Cisco IOS 软件版本 12.1(5)T 中已使用 Bug ID [CSCdr01713](http://www.cisco.com/cgi-bin/Support/Bugtool/onebug.pl?bugid=CSCdr01713)（[仅限注册用户](http://tools.cisco.com/RPF/register/register.do)）添加了该功能。

**注意：**如果未在虚拟模板接口中手动配置 IP MTU（使用上一部分中的选项），则仅自动调整 IP MTU。

最初，该功能在默认情况下处于启用状态，并且无法将其禁用。Cisco IOS 软件版本 12.2(3) 和 12.2(4)T 中的 Bug ID [CSCdt67753](http://www.cisco.com/cgi-bin/Support/Bugtool/onebug.pl?bugid=CSCdt67753)（[仅限注册用户](http://tools.cisco.com/RPF/register/register.do)）后来将命令 [no] ip mtu adjust 添加到 **vpdn-group** 下以启用和禁用该功能。默认情况下会启用该功能。该功能没有命令行界面 (CLI) 来更改仅面向 L2X 连接的默认值，这些连接不绑定到 **vpdn-group**（例如 SGBP 启动的 L2F 或 L2TP 隧道）。由于无法为多机箱多链路 PPP (MMPPP) 拓扑禁用该功能，再加上如下所述的 PMTUD 问题，引起许多用户投诉。出于这个原因，在 Cisco IOS 软件版本 12.2(6)、12.2(8)T 及更高版本中，开始使用 Bug ID [CSCdu69834](http://www.cisco.com/cgi-bin/Support/Bugtool/onebug.pl?bugid=CSCdu69834)（[仅限注册用户](http://tools.cisco.com/RPF/register/register.do)）将默认值更改为禁用 IP MTU 自动调整功能。

MTU 的手动和自动调整取决于终端主机之间的 PMTUD。虽然理论上行得通，但 PMTUD 不能在 Internet 中正常使用。有关 PMTUD 如何在 Internet 上中断的详细说明，请参阅 [RFC 2923](http://www.ietf.org/rfc/rfc2923.txt) 。[leavingcisco.com](http://www.cisco.com/swa/i/icon_popup_short.gif) [最大的问题是，“黑洞”的存在会导致网页下载在传输数据流的过程中显示为挂起。这些黑洞通常是由配置为滤掉 ICMP 消息的防火墙或路由器导致的。当大数据包的源不能从路由器接收“ICMP Host Unreachable”消息时，表示已超过 MTU，它不能减少数据包的大小。但是，它可以通过设置的 DF 位反复尝试并重新传输相同的数据包。由于这些数据包超过 PMTU 并且连接停止响应，因此会被 LNS 丢弃。](http://www.ietf.org/rfc/rfc2923.txt)

由于依赖 PMTUD 来检测 L2TP 隧道上的 IP MTU 较小存在问题，Cisco 已在 Cisco IOS 软件版本 12.2(4)T 中添加了 TCP 最大数据段大小 (MSS) 调整功能。

### 调整 TCP MSS

在 Cisco IOS 软件版本 12.2(4)T 及更高版本中，通过 Bug ID [CSCds69577](http://www.cisco.com/cgi-bin/Support/Bugtool/onebug.pl?bugid=CSCds69577)（[仅限注册用户](http://tools.cisco.com/RPF/register/register.do)）添加的 TCP 最大数据段大小调整功能允许路由器修改终端主机发送的传入和传出同步 (SYN) 数据包中所通告的 TCP MSS。通过将 TCP MSS 的值修改为低于通常的默认值 1460，可以避免将 TCP 作为最大尺寸的数据包的来源。应将 TCP MSS 调整为此类值，即包含 TCP/IP 报头并封装在 L2TP over UDP 中的 TCP 分段不超过输出接口的 IP MTU。TCP/IP 报头通常为 40 字节，L2TP over UDP 报头为另外的 40 字节。因此，通常 TCP MSS 应调整为 1420（1500 - 40 字节的 TCP/IP 报头 - 40 字节的 L2TP over UDP 报头）。

用于该任务的命令是 **ip tcp adjust-mss <mss>**，它是一个接口级别命令。

用于减少 L2TP 网络中的分段的最后一个选项要求为点对点协议客户端中的最大接收单元 (MRU) 协商提供支持。PPP 中的 MRU 选项允许对等体通告它的最大接收单元。例如，如果对等体通告 MRU 为 1460，则该对等体将不能处理有效负载超过 1460 字节的 PPP 帧。Cisco PPP 实施将接口的 MTU 用作它在 PPP 协商期间通告的 MRU 值。如果将 MTU 设置为默认的 1500 字节，则不会通告 MRU，因为这是 PPP 的标准默认值。但是，如果将 MTU 设置为 1460，则会通告 PPP MRU 为 1460 字节。如果 PPP 对等体侦听到在 PPP 协商期间通告的 MRU 并为该 PPP 链路调整其 MTU（并间接调整其 IP MTU），我们可以避免分段。通告 PPP MRU 为 1460 字节后，对等体应将 IP MTU 设置为 1460。这反过来会修改对等体在打开 TCP 连接时通告的 TCP MSS，并避免通过 L2TP 网络分段。

### 配置更小的 MTU

使用 **mtu <bytes>** 命令在虚拟模板接口中配置较低的 MTU。同样，这也需要支持 PPP 客户端以侦听在 PPP 协商期间通告的 MRU。已知的一个用于侦听 MRU 选项的客户端是 Windows XP PPP 客户端。但遗憾的是，其他通常部署的 PPP 客户端并不遵守它们应遵守的所通告的 PPP MRU。请参阅 PPP 客户端文档以确定是否将正确使用所通告的 PPP MRU。通过代理 LCP 运行 L2TP 时，由于在 LCP 阶段对 MRU 选项进行了协商，因此必须执行 LCP 重新协商。要启用 LCP 重新协商，请在 **vpdn-group** 下配置 **lcp renegotiation on-mismatch** 或 **lcp renegotiation always**。

降低 MTU 时，也会自动降低 IP MTU。当前，配置的 IP MTU 不能高于虚拟模板接口中的 MTU。可以通过 Bug ID CSCdx39828 将其作为一个功能/增强请求来跟踪（外部用户不能查看）。

该方法要求客户端在 LCP 协商期间侦听 MRU 选项。通常结合使用多个客户端：一些客户端侦听 MRU，而另一些不侦听。忽略 MRU 的客户端会导致如[自动调整 IP MTU](http://www.cisco.com/c/zh_cn/support/docs/dial-access/virtual-private-dialup-network-vpdn/24320-l2tp-mtu-tuning.html#automatic) 部分中介绍的 PMTUD 问题。对于这些客户端，您可以通过清除 IP 数据包内部的 DF 位有效禁用 PMTUD，从而可以利用不同的解决方法。您可以通过下列配置达到上述目的：

interface virtual-template1 ip policy route-map clear-df ! route-map clear-df permit 10 match ip address 101 set ip df 0 ! access-list 101 permit tcp any any

## 结论

Cisco IOS 软件提供了多种最大化 L2TP 交换性能的方法。PMTUD 是理想的解决方案。但是，由于 Internet 中的问题，有时它并不可靠。Cisco IOS 软件提供了一些备用机制以保持较高的 L2TP 交换性能并最大化用户连接。

# **案例分析8**

## MTU问题1

大家好，昨天碰到了一个用户故障申告，处理完了，总结一下，有些疑惑无法确认，请大家帮忙分析。

用户使用ADSL接入方式的VPDN,拓扑描述如下：

中心点SQL服务器---交换机---华为2631路由器（LNS）---光纤-------城域网交换机----城域网BAS（LAC）/华为8850----分点路由器华为2611--用户pc

用户分点采用2611进行pppoe拨号，拨通后用进行地址转换。分点pc机装有sql客户端软件。分支点路由器拨号后，用户pc能够访问中心点的sql服务器，ping通，但sql客户端无法连接中心点服务器。用ping -l 1500不通，逐步减少ping的报文长度，最后确认最大只能ping 1434长度的。

分析原因，由于用户采用pppoe方式的vpdn接入中心，用户发送的报文长度在局域网内mtu为1500 mss为1460，但由于到路由器上进行了pppoe的封装，用户发出报文外面加了pppoe的长度6byte+ppp长度 的2byte，对于用户端来说报文传送路径上最大传输单元就降低了8byte，局端bas与用户中心点LNS建立L2TP隧道，BAS终结用户报文的pppoe后，又封装进了L2tp报文中再封装到一层IP包中进行传输，到达用户中心路由器后解除l2tp的封装，转成正常以太网报文送向sql服务器。

看看用户ping中心服务器的icmp报文的封装是什么样的？

中间L2TP封装的结构如下：隧道ip层20byte+udp 8byte+L2TP 8byte+ppp 2 byte+ 隧道内 ip 20byte + icmp 8byte =======66byte 这就是用户为什么只能ping通1434以下长度的原因了，用户端sql客户端发起连接的报文长度大于了这个数据就无法正常传递了，但是我们知道ip报文传送过程中有其自适应网络传输介质的机制，也就是如果报文长度超过了接口mtu，并且报文df没有置位，这个报文将会分片传输，而到达对端后将进行报文的重组。如果报文中df位被置1，那么，用户的报文长度超过mtu后，设备会向报文原地址主机发出icmp type=3 code=4 的保文要求降低分组大小，主机降低到传送路径最低值发送报文，也就是pmtu。如果用户ip报文的 DF置1 ，而链路上某一台设备过滤了icmp type3 code4 的报文，用户报文遇到小mtu的路径后就不会收到响应来降低报文分组长度。通信就会受到影响。

基于这个思想，对用户连接网络的设备进行了逐一排查，并没有安全策略会挡住icmp报文，无可奈何的情况下，将用户分点路由器、中心点路由器的相关接口mtu强制成1380，这个数是我保守配置的，怕自己的知识不够，没能考虑到某一层协议的封装长度。呵呵。再保守一点，将用户的两端pc机和服务器的mtu也降到1380，呵呵。

故障解决，用户的ping -l 1500可以通了，SQL服务器也能够连接上了。

但是，为什么ip自动适应传输的机制没有正常响应呢？

通过在用户端路由器上出口sniffer ，发现了一个有点困惑的地方，分点用户主机发出的报文df置1了，而通过LNS进行L2TP封装的上层IP报文没有对DF位置1 ，那么，如果用户发出1500报文在本地路由器上就会受到icmp type 3 code 4 的应答，降低到1492报文长度+8个pppoe包头，到了BAS上又变成1492，但在L2tp tunnel 上由于要封装l2tp 包头，又需要将报文分成1434+20+8的长度进行封装 ，但又需要进行报文拆分，是向原主机发送 type3 code4的icmp报文呢？还是直接进行分片处理。这块就不太理解了，请大家给于分析解答，还有，在8850上 interface tunner 的mtu 1500 的值是否需要修改？

现象描述: 用户反馈ADSL用户使用VPN Client基于L2TP OVER IPSEC连接到内网后，访问内网服务器网页延迟较大，经常出现打开新WEB页面需要较长时间刷新的情况。

原因分析: 现场使用ping命令检测和WEB服务器的连通情况，发现无丢包情况，延时也在正常范围，网络正常。因此怀疑是数据包重传造成访问时长加大。

处理过程: 通过在防火墙抓包发现，每次从WEB服务器过来的数据包大小都为1500，再叠加L2TP和IPSEC包头后数据包均大于1500了，而网络中的网络设备MTU值一般都为1500，因此会造成网络中传输设备对此数据包进行分包处理，造成网络延时或重传，所以WEB页面打开延迟较大。

通过修改防火墙内网口的MTU值为1300，在叠加L2TP和IPSEC包头后数据包小于1500，网络恢复正常，ADSL用户访问内网服务器网页刷新正常。

## MTU问题2

 一、问题描述

L2tp组网用户反映，无法打开其内网的一个网站，地址为172.20.252.8，但可以打开其他网站，如：172.20.12.131；

 二、组网拓扑

 用户接入―交换机---城域网ERX（LAC） -----Eudemon300（LNS）----内部服务器

 三、问题处理及分析

 1、将用户割接至ME60，使用ME60做LAC，能正常访问网站172.20.252.8；与ERX做LAC现象不同；

 2、分别在正常和故障两种场景下，在用户端对拨号及访问网站的过程进行抓包对比，发现了问题：

 （1）服务器的回应报文进行了分片，正常情况下客户端可以收到分片后的三个报文（序号129,130,131）

 （2）ERX作LAC的时候，用户侧只抓到了分片后的第一个报文，丢弃了另外两个，导致无法访问网站（序号160）

 3、  Eudemon300上的VT3接口MTU是1500，服务器发送给客户端的包经过VT3接口处理并不会对其进行分片，而是直接加上38字节，即IP(20)+UDP(8)+ L2TP(6)+PPP(4，前加了“FF03”)；而Eudemon300上出LAC接口的MTU也是1500，那么此L2TP报文肯定会在对应端口上进行分片；ME60的版本及单板都支持L2TP报文的分片重组，怀疑ERX设备不支持L2TP报文的分片重组；

 4、  将Eudemon300上mss值改成1440，即LNS与用户端协商的发包大小改小，使之L2TP封装后字节数不超过Eudemon300上出LAC接口的MTU，L2TP报文不分片。问题解决。

## MTU问题3

MTU：Maximum Transmission Unit，是指一种通信协议的某一层上面所能通过的最大数据报大小（以字节为单位）。最大传输单元这个参数通常与通信接口有关（网络接口卡、串口等）。在现网中经常会遇到一些网络是通的，但是部分网页打不开的情况；或则是在有VPN技术的情况下，隧道是建立的但是某些应用程序没有办法使用；再则是比较明显的小包能够Ping得通，但是大于一定的包就Ping不同。借此机会讨论讨论关于设备MTU的事情，欢迎各位踊跃参与！！  
　　先从一个实验说起  
　　实验拓扑如下：  
　　PC1-------VPN网关1―――――VPN网关2--------PC2  
　　在上述组网中，VPN网关1和2之间建立了一条GRE隧道，PC1与PC2通过该GRE隧道互访。  
　　这时，我们会发现，PC1在ping 1448的报文时，在PC2上抓包发现没有被分片，而ping 1449时，PC2上会发现PC1过来的报文已经被分片了。为什么呢？  
　　PC1出来的IP报文长度为1448＋8（ICMP报头长）＋20（IP报头长），到达VPN网关1，VPN网关1发到GRE隧道口，封装GRE头（4字节），再加上外层IP头，到达VPN网关外层以太口，这时IP报文的长度已经变为：1448＋8＋20＋4＋20＝1500字节，刚好等于以太口的MTU，于是被顺利传送。而ping 1449时，到达外层以太口为1501字节，超出了1500的MTU，又因为报文DF位未置1，即可以分片，VPN网关于是将该报文分片发送出去。这就是我们所看到的现象。  
　　在上述实验中，由于应用程序是ping，所以报文可以被分片，因此互通没有问题。但是如果是WEB访问等应用，则有些报文是不允许分片的，这样在外层以太口就会将超过1500的报文丢掉，导致无法通讯。  
　　从上述实验可以看到，由于VPN会额外加入一些报文头，如果通讯双方的MTU不能随之改变的话，就容易产生不通的问题。  
　　下面以HTTP为例，说明为何产生此问题并如何解决。  
　　先看看HTTP为何无法像ICMP那样自动分片通讯。  
　　假设PC1/2建立了HTTP连接后，PC2希望从PC1下载一个大的网页。PC2开始发送，其IP的DF位置1，不允许分片，IP报文长度为1500字节。到达VPN网关1的外网口后，VPN网关1发现其长度超过了1500个字节，于是将其丢弃，并给PC1发回一个目的地址不可达的ICMP信息，出错代码为”Fragmentation needed”，表示需要分片，但不允许分片，同时给出”MTU of next hop: 1500”。PC1接收到该消息后，又按照1500字节对外发送，又被丢弃，于是就形成了循环，无法通讯。  
　　根据上述的分析，很容易得到如下解决方式，在VPN网关1的出接口设置MTU为1500－4－20＝1476，这样VPN网关1返回ICMP不可达消息时将给出”MTU of next hop: 1476”。PC2将以1476作为自己的最大MTU对外发送，到达VPN网关1，封装GRE和外层IP头后就不会超过1500而顺利发到对端。  
　　这时仅解决了下载的问题，如果PC2需要将大文件上传到PC1，同样需要设置VPN网关2的出接口MTU值小于1476。  
　　当然，还可以更改VPN网关1的出接口的TCP MSS数值，将其更改为1500-4-20-20（TCP头）=1456字节，也可保证HTTP等TCP应用顺利通过。但该情况仅适用于TCP应用。  
　　上述解决方式同样适用于其他隧道技术，在L2TP、IPSEC等应用时可以相应的根据其包头数值设置MTU或MSS  
　　**总结：**  
　　1、GRE隧道会增加 GRE报文头字节+外层IP报文头20字节  
　　2、收集了一些常见应用的MTU值，希望对大家有帮助  
　　1500. 以太网信息包最大值，也是默认值。是没有PPPoE和VPN 的网络连接的典型设置。是各种路由器、网络适配器和交换机的默认设置  
　　1492. PPPoE 的最佳值  
　　1472. 使用 ping 的最大值 （大于此值的信息包会先被分解）  
　　1468. DHCP的最佳值  
　　1430. VPN和PPTP 的最佳值  
　　576. 拨号连接到ISP的标准值

## PPTP/L2TP over PPPoE的準確MTU/MRU值

|  |
| --- |
| Ethernet MinSize = 512bit = 64 Byte |

|  |  |
| --- | --- |
| 2 | Ethernet MaxSize = 1518 Byte |

|  |  |
| --- | --- |
| 3 | so Ethernet IP MTU = 1518 - 18 ( 6 SRCMAC+ 6 DSTMAC+ 2 TYPE+ 4 CRC) = 1500 B |

|  |  |
| --- | --- |
| 4 | so Ethernet IP TCP MSS = 1500 - 40 ( 20 IP\_HEADER + 20 TCP\_HEADER) = 1460 B |

|  |  |
| --- | --- |
| 5 | so Ethernet IP UDP MTU/MRU = 1500 - 28 ( 20 IP\_HEADER + 8 UDP\_HEADER ) = 1472 B |

|  |  |
| --- | --- |
| 6 | so PPPoE MTU/MRU = 1500 - 8 ( 6 PPPoE\_SESSION + 2 PPP\_HEADER ) = 1492 B |

|  |  |
| --- | --- |
| 7 | so TCP over PPPoE MSS = 1492 ( PPPoE MTU/MRU ) - 40 ( 20 IP\_HEADER + 20 TCP\_HEADER) = 1452 |

|  |  |
| --- | --- |
| 8 | so PPTP MTU/MRU = 1500 - 56 ( 20 IP\_HEADER + 20 TCP\_HEADER + 12 GRE\_HEADER + 4 PPP\_HEADER ) = 1444 B |

|  |  |
| --- | --- |
| 9 | so TCP over PPTP MSS = 1444 ( PPTP MTU/MRU ) - 40 ( 20 IP\_HEADER + 20 TCP\_HEADER) = 1404 |

|  |  |
| --- | --- |
| 10 | so L2TP MTU/MRU = 1500 - 40 ( 20 IP\_HEADER +8 UDP\_HEADER + 8 L2TP\_HEADER + 4 PPP\_HEADER ) = 1460 B |

|  |  |
| --- | --- |
| 11 | so TCP over L2TP MSS = 1460 ( L2TP MTU/MRU ) - 40 ( 20 IP\_HEADER + 20 TCP\_HEADER) = 1420 B |

|  |  |
| --- | --- |
| 12 | so PPTP over PPPoE MTU/MRU = 1492 ( PPPoE MTU/MRU ) - 56 ( 20 IP\_HEADER + 20 TCP\_HEADER + 12 GRE\_HEADER + 4 PPP\_HEADER ) = 1436 B |

|  |  |
| --- | --- |
| 13 | so PPTP over PPTP MTU/MRU = 1444 ( PPTP MTU/MRU ) - 56 ( 20 IP\_HEADER + 20 TCP\_HEADER + 12 GRE\_HEADER + 4 PPP\_HEADER ) = 1388 B |

|  |  |
| --- | --- |
| 14 | so PPTP over L2TP MTU/MRU = 1460 ( L2TP MTU/MRU ) - 56 ( 20 IP\_HEADER + 20 TCP\_HEADER + 12 GRE\_HEADER + 4 PPP\_HEADER ) = 1404 B |

|  |  |
| --- | --- |
| 15 | so L2TP over PPPoE MTU/MRU = 1492 ( PPPoE MTU/MRU ) - 40 ( 20 IP\_HEADER +8 UDP\_HEADER + 8 L2TP\_HEADER + 4 PPP\_HEADER ) = 1452 B |

|  |  |
| --- | --- |
| 16 | so L2TP over PPTP MTU/MRU = 1444 ( PPTP MTU/MRU ) - 40 ( 20 IP\_HEADER +8 UDP\_HEADER + 8 L2TP\_HEADER + 4 PPP\_HEADER ) = 1404 B |

|  |  |
| --- | --- |
| 17 | so L2TP over L2TP MTU/MRU = 1460 ( L2TP MTU/MRU ) - 40 ( 20 IP\_HEADER +8 UDP\_HEADER + 8 L2TP\_HEADER + 4 PPP\_HEADER ) = 1420 B |

故而，PPTP over PPPoE的準確MTU值是 1436，L2TP over PPPoE的準確MTU是1452。

別人說的 1400 偏保守了點。

如果你用 ADSL 上網，然後用 PPTP 來翻Wall，那麼，你實際上是 PPP 協議跑在 TCP 協議上再跑在 IP 協議上再跑在 PPP 協議上再跑在 IP 協議上再跑在以太網協議上。1518 字節的最大以太網 frame，扣來扣去，就剩下 1436。同理L2TP跑在PPP鏈路上，扣來扣去也就只剩1452B了，難怪筆者之前VPN上不去網，原來是全把數據包丟掉了。

如果讀者看明白了上面的解釋，那麼考慮你用 pptp 連公司的 vpn，公司又 pppoe(adsl撥號)，然後你再 pptp 來翻功夫網，那麼，你的 MTU 只能設為 1518-18-8-56-56=1380 字節。很簡單吧？

## MTU MSS PPPoE http超时的问题

 (2013-07-16 15:29:26)

[[http://simg.sinajs.cn/blog7style/images/common/sg_trans.gif](javascript:;)转载▼](javascript:;)

|  |  |
| --- | --- |
| 标签： [mtu](http://search.sina.com.cn/?c=blog&q=mtu&by=tag)  [mss](http://search.sina.com.cn/?c=blog&q=mss&by=tag) | 分类： [数通](http://blog.sina.com.cn/s/articlelist_1114261893_1_1.html) |

 **名词**

    MTU: Maxitum Transmission Unit 最大传输单元

    MSS: Maxitum Segment Size 最大分段大小

    PPPoE: PPP Over Ethernet（在以太网上承载PPP[协议](http://cisco.chinaitlab.com/List_11.html)）

**PPPoE拔入连接中的相关问题**

    ADSL 的PPPoE连接的MTU是1492, 而Widows和Linux缺省的MTU是1500, 当浏览器相WEB[服务器](http://server.chinaitlab.com/)发送请求时， 要求MSS是1460, 一些站点屏蔽了ICMP3/4访问或者禁用了PMTUD[协议](http://cisco.chinaitlab.com/List_11.html)而导致MSS值是1460, 当WEB[服务器](http://server.chinaitlab.com/)返回的包小于1452时，客户端可以正常接受数据， 超过1452时候无法得到数据。[说得有点乱，也就是说在pppoe时，MSS超过1452会出问题。]

    用户请求一个网页时，个人计算机 与放置网页的Web服务器之间进行客户机/服务器协商，以确定一个最大的MTU尺寸。进行协商的个人计算机的默认 MTU大小为1500个字节（Windwos/index.html' target='\_blank'>[Windows](http://windows.chinaitlab.com/) 3x, 9x, NT,ME等），所以Web服务器经协商确定的MTU大小为1500字节。因此，无论您的[路由](http://cisco.chinaitlab.com/List_6.html)器MTU大小配置为多少，Web服务器所发送的数据包最大字 节数仍为1500.

    一些网页不能完全[下载](http://download.chinaitlab.com/)的原因是：如果个人计算机MTU配置错误，发送给路由器的数据包超过1492个字节，那么路由器将对IP数据包进行分段。这种数据分段不在通过通用访问集中器（UAC）的 返回路径上进行（[Cisco](http://cisco.chinaitlab.com/) 6400 or 7200）。UAC将丢弃接收到的超过1490字节的数据包，同时生成[互联网](http://internet.chinaitlab.com/)控制信息[协议](http://cisco.chinaitlab.com/List_11.html)（ICMP）消息并发送给发送超过1490字节数据包的Web服 务器。ICMP通知该Web服务器其所发送了一个超大的数据包，需要重发一个小一点MTU值的数据包

    出现这样的问题是因为许多Web服务器阻止了ICMP消息，因为它会导致服务器不断发送1500字节的数据包。这些数据包被丢弃，所请求的Web页因此不能[下载](http://download.chinaitlab.com/)。如果Web服务器配置正确而且可以接收ICMP消息，服务器应调整其MTU大小，直到页面完全[下载](http://download.chinaitlab.com/)。

    Web服务器开始发送的数据包少于1492字节的最大值，然后再发送超过这个最大值的数据包时就会出现了只能下载一部分网页的情况。服务器不断发送超过最大值的数据包造成网页仅能下载一部分，并且在状态条里显示 "等待回应……" 信息

 **相关参数介绍**

    [MTU]

    先 说说这MTU最大传输单元，这个最大传输单元实际上和链路层协议有着密切的关系，让我们先仔细回忆一下EthernetII帧的结构 6DMAC+6SMAC+2Type+Data+4CRC 由于以太网传输电气方面的限制，每个以太网帧都有最小的大小64bytes最大不能超过1518bytes,对于小于或者大于这个限制的以太网帧我们都可 以视之为错误的数据帧，一般的以太网转发设备会丢弃这些数据帧。

    （注：小于64Bytes的数据帧一般是由于以太网冲突产生的"碎片"或者线路干扰或者坏的以太网接口产生的，对于大于1518Bytes的数据帧我们一般把它叫做Giant帧，这种一般是由于线路干扰或者坏的以太网口产生）

    由 于以太网EthernetII最大的数据帧是1518Bytes这样，刨去以太网帧的帧头（DMAC目的MAC地址 48bit=6Bytes+SMAC源MAC地址48bit=6Bytes+Type域2bytes）14Bytes和帧尾CRC校验部分 4Bytes（这个部门有时候大家也把它叫做FCS），那么剩下承载上层协议的地方也就是Data域最大就只能有1500Bytes这个值我们就把它称之 为MTU.这个就是网络层协议非常关心的地方，因为网络层协议比如IP协议会根据这个值来决定是否把上层传下来的数据进行分片。

    当两台远程PC互联的时候，它们的数据需要穿过很多的路由器和各种各样的网络媒介才能到达对端，网络中不同媒介的MTU各不相同，就好比一长段的水管，由不同粗细的水管组成（MTU不同 ）通过这段水管最大水量就要由中间最细的水管决定。

    对 于网络层的上层协议而言（我们以TCP/IP协议族为例）它们对水管粗细不在意它们认为这个是网络层的事情。网络层IP协议会检查每个从上层协议下来的数 据包的大小，并根据本机MTU的大小决定是否作"分片"处理。分片最大的坏处就是降低了传输性能，本来一次可以搞定的事情，分成多次搞定，所以在网络层更 高一层（就是传输层）的实现中往往会对此加以注意！有些高层因为某些原因就会要求我这个面包不能切片，我要完整地面包，所以会在IP数据包包头里面加上一 个标签：DF（Donot Fragment）。这样当这个IP数据包在一大段网络（水管里面）传输的时候，如果遇到MTU小于IP数据包的情况，转发设备就会根据要求丢弃这个数据 包。然后返回一个错误信息给发送者。这样往往会造成某些通讯上的问题，不过幸运的是大部分网络链路都是MTU1500或者大于1500.

 对于UDP协议而言，这个协议本身是无连接的协议，对数据包的到达顺序以及是否正确到达不甚关心，所以一般UDP应用对分片没有特殊要求。

    对于TCP协议而言就不一样了，这个协议是面向连接的协议，对于TCP协议而言它非常在意数据包的到达顺序以及是否传输中有错误发生。所以有些TCP应用对分片有要求---不能分片（DF）。

    [PPPoE]

    所谓PPPoE就是在以太网上面跑PPP协议，有人奇怪了，PPP协议和Ethernet不都是链路层协议吗？怎么一个链路层跑到另外一个链路层上面去了，难道升级成网络层协议了不成。其实这是个误区：就是某层协议只能承载更上一层协议。

    为什么会产生这种奇怪的需求呢？这是因为随着宽带接入（这种宽带接入一般为Cable Modem或者xDSL或者以太网的接入）由于以太网缺乏认证计费机制而传统运营商是通过PPP协议来对拨号等接入服务进行认证计费的，所以就出了这么一个怪胎：PPPoE.

    PPPoE带来了好处，也带来了一些坏处，比如：二次封装耗费资源，降低了传输效能等等，这些坏处俺也不多说了，最大的坏处就是PPPoE导致MTU变小了以太网的MTU是1500,再减去PPP的包头包尾的开销（8Bytes），就变成1492.

    如果两台主机之间的某段网络使用了PPPoE那么就会导致某些不能分片的应用无法通讯。

    这个时候就需要我们调整一下主机的MTU,通过降低主机的MTU,这样我们就能够顺利地进行通讯了。

    当然对于TCP应用而言还有另外的解决方案。

    [MSS]

    MSS最大传输大小的缩写，是TCP协议里面的一个概念。

    MSS 就是TCP数据包每次能够传输的最大数据分段。为了达到最佳的传输效能TCP协议在建立连接的时候通常要协商双方的MSS值，这个值TCP协议在实现的时 候往往用MTU值代替（需要减去IP数据包包头的大小20Bytes和TCP数据段的包头20Bytes）所以往往MSS为1460.通讯双方会根据双方 提供的MSS值得最小值确定为这次连接的最大MSS值。

 **有关的一些解决办法**

    1. 使用ping找到你的MAX MTU Size

    ping -l mtu\_size-28 -f your\_gateway

    从1500开始，知道没有收到错误信息the Packet needs to be fragmented but DF set. 28是IP和ICMP协议所用的字节。一般而言PPPoE是1492.

    上面这个ping命令是win下的写法，[linux](http://linux.chinaitlab.com/)中的ping的命令参数有所不同，但有相应的功能参数，-s指定包大小， -M指定分片的要求。

    2.调整PC的MTU

[Windows](http://windows.chinaitlab.com/) 2000/XP

    注册表：HKEY\_LOCAL\_MACHINE SYSTEM CurrentControlSet Services

    TcpipParameters Interfaces, and （Edit -> New -> DWORD Value），

    and name the value MTU.

    Double-click the new value, choose the Decimal option, and type the MTU value determined above. 重启。

    Windows 98

    注册表： HKEY\_LOCAL\_MACHINE System CurrentControlSet Services ClassNet 增加MaxMTU string项，

    Linux

    ifconfig interface mtu mtu\_size

3. 调整[linux](http://linux.chinaitlab.com/)网关

    Iptables

    1）增加规则：

    iptables -o ppp\_interface -I FORWARD 1 -p tcp --tcp-flags SYN,RST SYN -m tcpmss --mss 1400:1536 -j TCPMSS --clamp-mss-to-pmtu

    出站包大小改为pppx的MTU尺寸。

    2）或者 clamp mss size in pppoe

    RedHat中修改/etc/sysconfig/networking-script/ifcfg-ppp0

    CLAMPMSS=1452

    Debian中修改 /etc/ppp/peers/dsl-provider

    增加

    pty "/usr/sbin/pppoe -I eth1 -T 80 -m 1452

    注释掉 plugin, rp-pppoe.so 好像有bug

    4调整cisoc网关

    如 果我们在中间路由器上把每次TCP连接的最大MSS进行调整这样使得通过PPPoE链路的最大MSS 值加上数据包头包尾不会超过PPPoE的MTU大小1492这样就不会造成无法通讯的问题。所以上面的问题可以通过ip tcp adjust-mss 1452来解决。

    这条命令要在IOS12以上的版本中才有。

    当然问题也可以通过修改PC机的MTU来解决。

    4.cisco的相关解说

[Cisco](http://cisco.chinaitlab.com/)的TCP Adjust MSS Feature:

    The TCP MSS Adjustment feature enables the configuration of the

    maximum segment size （MSS） for transient packets that traverse a router,

    specifically TCP segments in the SYN bit set, when Point to Point Protocol

    over Ethernet （PPPoE） is being used in the network. PPPoE truncates the

    Ethernet maximum transmission unit （MTU） 1492, and if the effective MTU

    on the hosts （PCs） is not changed, the router in between the host and the

    server can terminate the TCP sessions. The ip tcp adjust-mss command

    specifies the MSS value on the intermediate router of the SYN packets to

    avoid truncation.

    其它网络地址：<http://www.yuanma.org/data/2008/0116/article_2946.htm>

<http://ewangsoft.blog.163.com/blog/static/7721558220093200840864/>

    5.windows server 2003 的相关解说

    这段话源自：《Windows Server 2003 技术内幕（提高篇）》（ISBN 7-302-10332-1）

    P471

    PPPoE 所施加的一个重要的限制就是消减了客户端能传输的最大帧大小。标准的Ethernet帧具有1500字节的有效负载荷，这个参数称为"最大传输单元" （Maximum Transmission Unit,MTU）。相反，由于PPPoE帧的header要占用6字节，还要用另外2个字节位来设置PPP协议ID,所以它的MTU为1492字节。

    TCP/IP 客户端协商建立的Web服务器的一个连接时，两个实体将协商一个MTU.这在正常情况下是1500字节。但是，假如客户端在一个使用 PPPoE连接的服务器后面（该服务器在DSL服务器提供商处），PPPoE服务器就会丢弃帧，并向Web服务器返回一条"Internet Control Management Protocol     " （ICMP）消息，告诉它的MTU是1492.加入Web服务器被配置成拒绝ICMP（大多Web服务器都会这样设置），网页就无法正确地加载。

    为了查实这种情况，一个好用的诊断技术是从客户端上对Web服务器执行Ping操作，并用 -l 开关了指定一个载荷大小，再用一个 -f 开关来禁止分割数据库。具体语法是：

    ping -f -l 1492

    如果屏幕上返回一条 "Packet needs to be fragmented but DF set"     错误提示，你就知道已经超过了web服务器或者沿途的某个路由器的MTU设置。

    为 了避免产生此问题，你需要在位于PPPoE服务器背后的客户端上减小MTU值。为此，你既可以在每个客户端上进行设置，也可以在一个客户端上进行设置，然 后使用Regedit将修改导出至一个文件，并采取一个登录脚本的方式，将文件应用于其他客户端。你需要改动的地方是：

    项：HKLM | System | CurrentControlSet | Services | Ndiswan | Paremeters | Protocols | 0

    值：ProtocolType

    数据：0x0800 （REG\_DWORD）

    值：PPPProtocolType

    数据：0x0021（REG\_DWORD）

    值：ProtocolMTU

    数据： （REG\_DWORD）

## ppp协议 详细说明

 1. 两台设备使用PPP连接，直连接口属于不同网段，是否可以ping通？

可以ping通。因为PPP在进行IPCP协商后会把学习到的直连路由加入到路由表中。

2. PPP 协议位于TCP/IP协议栈中哪一层？其主要功能是什么？  
PPP协议处于TCP/IP协议栈中的链路层（第二层），是一种提供在点到点链路上传输封装网络层数据包的数据链路层协议；PPP的主要功能是用来在支持全双工的同异步链路上进行点到点之间的数据传输。

3. PPP 主要由哪几类协议组成？各类协议的功能是什么？  
PPP定义了一整套的协议，其组成及功能如下：

(1)        链路控制协议（LCP：Link Control Protocol），主要用来建立、拆除和监控数据链路。

(2)        网络层控制协议（NCP：Network Control Protocol），主要用来协商在该数据链路上所传输的数据包的格式与类型。

(3)        扩展协议族，主要用于提供对PPP功能的进一步支持。例如用于网络安全方面的验证的PAP和CHAP。

4. PPP 协议有哪些特点？  
PPP协议的特点如下：

(1)        PPP协议与其他链路层协议不同，PPP既支持同步链路又支持异步链路，而如X25、FR 等数据链路层协议只对同步链路提供支持；

(2)        PPP协议具有各种NCP协议，如IPCP、IPCPv6、IPXCP协议，它更好地支持了网络层协议；

(3)        PPP协议具有验证协议PAP、CHAP，它更好了保证了网络的安全性；

(4)        PPP协议更容易扩充。

5. LCP 协商哪些内容，过程如何？  
      PPP在建立链路之前要进行一系列的协商，PPP 首先进行LCP协商，协商内容包括MRU（最大传输单元）、魔术字（magic number）、 验证方式异步字符映射等选项（详见RFC1661）；LCP协商成功后才进入Establish(链路建立)阶段；任何阶段的协商失败都将导致链路的拆除，MRU用于协商PPP链路的最大包传输能力，该选项向对端指出本端能够接收的最大报文长度，它以字节为单位，指出Information+Padding的长度，不包括framing、Protocol、FCS和任何为透明传输所填充的比特或字节。该选项的缺省值为1500字节。该选项实际上指出本端的一种能力，但并不要求对端一定要最大化使用这种能力。

例如，本端在配置选项中声明MRU为2048 字节，这并不是说要求对端发送的任何报文都要达到2048字节，若对端能满足MRU选项，则回Config ACK报文，若不满足，则回Config NAK报文；魔术字主要用于检测链路自环，PPP靠发送Echo Request和Echo Reply报文来检测自环和维护链路状态；如连续发现有超过最大自环允许数目，Echo Request报文中魔术字与上次发送魔术字相同，则判定网络发生自环现象。如链路发生自环，则就需采取相应措施对链路复位；另外，LCP发送Config Request时也可以检测自环，LCP发现自环后，在发送一定数目的报文后也会复位链路，如果PPP发送的Echo Request报文产生丢失，则在连续丢失最大允许丢失的个数之后，将链路复位，以免过多的无效数据传输。异步字符映射用于同异步转换。

6. PAP与CHAP的验证过程分别是由哪方发起的？PAP和CHAP的最大不同点是什么？  
PAP是首先由被验证方将自己的用户名及密码送给验证方；而CHAP验证是首先由验证方发起；验证过程的主要区别为：PAP为明文传送密码，而在CHAP验证过程中密码是不在线传送的，属于密文传送。

7. PAP的详细验证过程是什么样的？  
PAP为两次握手协议，详细验证过程如下：当两端链路可相互传输数据时，被验证方发送本端的用户名及口令到验证方，验证方根据本端的用户表（或Radius服务器）查看是否有此用户，口令是否正确。如正确则会给对端发送ACK报文，通告对端已被允许进入下一阶段协商；否则发送NAK报文，通告对端验证失败。一次认证失败并不会直接将链路关闭，只有当验证不过次数达到一定值时，才会关闭链路，来防止因误传、网络干扰等造成不必要的LCP重新协商过程。

8. CHAP的详细验证过程是什么样的？  
CHAP为三次握手协议，其单向验证过程分为两种情况：验证方配置了用户名和验证方没有配置用户名。

l              验证方配置了用户名的CHAP验证过程如下：

(1)        验证方主动发起验证请求，验证方向被验证方发送一些随机产生的报文（Challenge），并同时将本端的用户名附带上一起发送给被验证方；

(2)        被验证方接到验证方的验证请求后，根据此报文中验证方的用户名在本端的用户表查找该用户对应的密码，如果在用户表找到了与验证方用户名相同的用户，便利用报文ID、此用户的密钥（密码）和MD5算法对该随机报文进行加密，将生成的密文和被验证方自己的用户名发回验证方（Response）；

(3)        验证方用自己保存的被验证方密码和MD5算法对原随机报文加密，比较二者的密文，根据比较结果返回不同的响应（Acknowledge or Not Acknowledge）。

l              验证方没有配置用户名的CHAP验证过程如下：

(1)        验证方主动发起验证请求，验证方向被验证方发送一些随机产生的报文（Challenge）；

(2)        被验证方接到验证方的验证请求后，利用报文ID、配置的CHAP密码和MD5算法对该随机报文进行加密，将生成的密文和自己的用户名发回验证方（Response）；

(3)        验证方用自己保存的被验证方密码和MD5算法对原随机报文加密，比较二者的密文，根据比较结果返回不同的响应（Acknowledge or Not Acknowledge）。

9. PPP LCP协商中的MRU选项是如何取值的？  
     PPP LCP协商的MRU（Maximum-Receive-Unit 最大接收单元）是针对PPP报文而言，指发送端PPP报文中信息域的长度不能超过MRU，否则接收端不负责处理，可能会将报文丢弃。

大多数厂家不提供MRU的设置命令而是把MTU的配置值作为本端MRU的值与对端协商MRU。最终PPP链路的MTU的取值应该是：MIN（配置MTU，MIN（本端MRU，对端MRU））

因此，封装PPP的接口上的MTU值被改变后，如想让这个改变立即生效，就需触发LCP重新协商。

这里描述的是普通PPP的情形，MP、PPPOX等PPP应用的MTU的最终取值更加复杂一些。

MRU选项是LCP协商成功的必要选项，这个选项最终必须协商成功，不能忽略，最终PPP双方必须确认这个选项，不能否认或拒绝。

10. PPP LCP协商中的ACCM选项有什么意义？  
     ACCM（Async-Control-Character-Map 异步控制字符映射）选项是在异步链路上用来通知对端哪些字符被本端用于控制，必须被转义（映射）。

     PPP帧中FLAG字段的值是0X7E（实际上大多链路层协议帧中FLAG字段都是这个值）。当这个值出现在帧内容中时需对它进行转义，同步链路中，这个过程是通过位比特填充的技术完成的（即在第5个1与第6个1之间填充0）。异步链路中，特殊字符0X7D被用做转义字符，对一些字符进行转义。由于异步链路中，有些串行接口驱动程序或MODEM需要用一些字符作为控制字符使用，字符的范围是0X00－0X1F（0-31）所以需要对这些字符进行转义处理，大多数情况下并不是这个范围内的所有字符都作为控制字符，因此PPP LCP提供这个选项用来告诉对端，本端将哪些字符作为控制字符处理，请对端将这些字符转义后发送。

ACCM选项用一个32位整数的位的值来表示哪些字符用作控制字符，譬如0X000A0000，第17位与第19位为1，说明字符17（XON）与字符19（XOFF）作为控制字符使用，需要进行转义。

在异步链路上，尤其是低带宽、高时延的异步线路（例如无线）上合理设置ACCM值可以有效避免在数据传输过程中引入额外时延。

异步链路上的LCP协商，这个选项是必要选项，必须协商成功。

11. PPP LCP协商中的Magic-number选项有什么意义？  
Magic-number选项用来协商双方的魔术字，两端魔术字不能重复，魔术字可用来检测链路的环回情况。

这是一个必要选项，必须被确认否则LCP不能UP。

12. PPP LCP协商中的PFC选项有什么意义？  
PFC（Protocol-Field-Compression 协议域压缩）选项是用来协商PPP报文的协议域是否可以被压缩，用高位字节的最低位表示此报文是否协议域被压缩，1表示是，0表示没有。压缩后协议域的两个字节会被压缩成一个字节，如IP报文中的0021可被压缩为21。实施者必须能够接收这两种情况（压缩或没有被压缩）的报文。

13. PPP LCP协商中的ACFC选项有什么意义？  
ACFC（Address-and-Control-Field-Compression地址控制域压缩）选项是用来协商PPP报文的地址、控制域是否可以被压缩。压缩后地址域和控制域会被省略。LCP报文不能进行地址控制域压缩。

14. MP LCP协商中的MRRU选项有什么意义？  
MRRU（Multilink Maximum Received Reconstructed Unit MP最大接收重构单元）选项可用来通知对端本端要实施MP，这个选项的值是指接收端重组后的MP报文的最大值。这个选项的确认与拒绝主要用来确定对端是否支持MP链路的使用。该选项被拒绝后，则该通道不能作为MP链路使用，而只能作为普通PPP链路使用。

15. MP链路上的MTU是如何计算的？  
一般MP链路上的MTU的取值依据下面的公式：

MINMRU=MIN（本端MRU，对端MRU）

MINMRRU=MIN（本端MRRU，对端MRRU）

MTU=MIN(配置MTU，MIN（MINMRU，MINMRRU）)  
如果MP通道是PPPoE之类的链路，MP MTU还要考虑底层的MTU，会更复杂一些。

# ****L2TP协议简介****

# 一、L2TP实现的两种方式

**名词解释：**

**LAC(L2TP Access Concentrator L2TP访问集中器)**

       是附属在交换网络上的具有PPP端系统和L2TP协议处理能力的设备。LAC一般是一个网络接入服务器NAS，主要用于通过PSTN/ISDN网络为用户提供接入服务。

**LNS(L2TP Network Server L2TP网络服务器）**

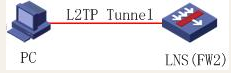
       是PPP端系统上用于处理L2TP协议服务器端部分的设备。

**VPDN（Virtual Private Dial Network，虚拟私有拨号网）**

        指利用公共网络（如ISDN和PSTN）的拨号功能及接入网来实现虚拟专用网。

注：如下示例中使用的防火墙（Firewall，FW）既可以当做LAC也可以作为LNS使用。

### 1、PC直接拨号到LNS，组网如图1所示



### 2、PC通过LAC拨号连接到LNS，组网如图2所示



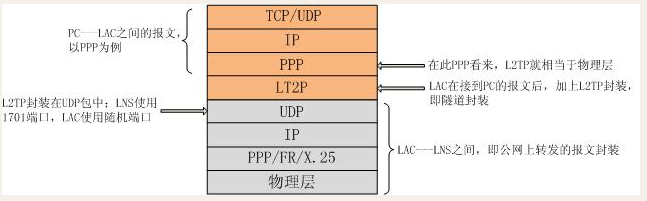
    因为一般都使用用PC---LAC---LNS组网，且此种组网包含了PC---LNS的组网形态，故后续描述均已PC---LAC---LNS为例。

        LAC位于LNS和主机之间，用于在LNS和主机之间传递信息包，把从主机收到的信息包按照L2TP协议进行封装并送往LNS，将从LNS收到的信息包进行解封装并送往远端系统。LAC与主机之间可以采用本地连接或PPP链路，VPDN应用中通常为PPP链路。LNS作为L2TP隧道的另一侧端点，是LAC的对端设备，是被LAC进行隧道传输的PPP会话的逻辑终止端点。

# 二、L2TP封装位置分析

       如下图3所示，从图中至上而下的分析，为PC的报文在PPP内网环境中发送到LAC，由LAC封装L2TP，再通过外网的报文正常转发给LNS的报文封装过程。

注：设计的网络环境为PC---LAC为内网使用PPP协议，LAC---LNS为外网使用协议由服务供应商自定。



# ****三、理解L2TP几个重要的概念****

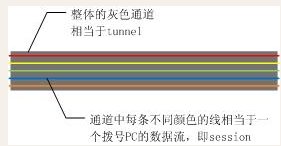
### ****1、隧道和会话的概念****

          在一个LNS和LAC对之间存在着两种类型的连接，一种是隧道（Tunnel）连接，一对LAC和LNS中可以有多个L2TP隧道；另一种是会话（Session）连接，它复用在隧道连接之上，用于表示承载在隧道连接中的每个PPP会话过程。

        隧道由一个控制连接和一个或多个会话（Session）组成。会话连接必须在隧道建立（包括身份保护、L2TP版本、帧类型、硬件传输类型等信息的交换）成功之后进行，每个会话连接对应于LAC和LNS之间的一个PPP数据流。控制消息和PPP数据报文都在隧道上传输。

* L2TP使用Hello报文来检测隧道的连通性。LAC和LNS定时向对端发送Hello报文，若在一段时间内未收到Hello报文的应答，该隧道连接将被断开。
* L2TP报文头中包含隧道标识符（Tunnel ID）和会话标识符（Session ID）信息，用来标识不同的隧道和会话。隧道标识相同、会话标识不同的报文将被复用在一个隧道上，报文头中的隧道标识符与会话标识符由对端分配。

        隧道（tunnel）和会话（session）的关系，如图4所示；可以形象的理解为会话是建立在隧道之中的，隧道想成一个有10个车道的高速公路，一台拨号PC的数据流为一个会话，相当于占用了一个车道（告诉公路有多少车道是设备规定好的），这个车道只能跑这个运载这个PC的报文的卡车。（比如某型号设备每条隧道最多支持1000个会话）。



### ****2、控制消息和数据消息的概念****

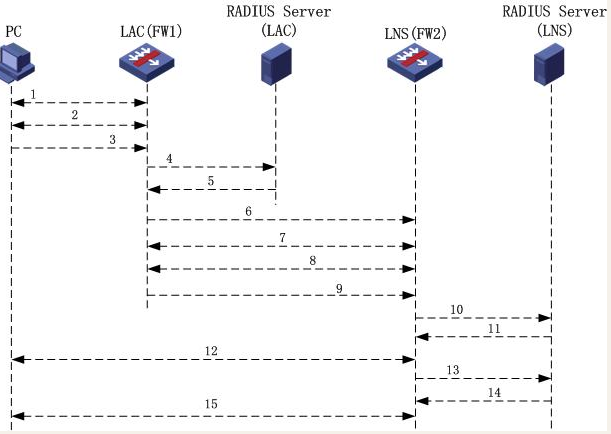
  L2TP中存在两种消息：控制消息和数据消息。

* **控制消息：**用于隧道和会话连接的建立、维护以及传输控制；控制消息的传输是可靠传输，并且支持对控制消息的流量控制和拥塞控制。
* **数据消息：**用于封装PPP帧并在隧道上传输；数据消息的传输是不可靠传输，若数据报文丢失，不予重传，不支持对数据消息的流量控制和拥塞控制。

  控制消息和数据消息共享相同的报文头。

# 四、L2TP隧道的呼叫建立流程

### ****1、L2TP隧道的呼叫建立流程****



(1) 用户端PC机发起呼叫连接请求；

(2) PC机和LAC端进行PPP LCP协商；

(3) LAC对PC机提供的用户信息进行PAP或CHAP认证；

(4) LAC将认证信息（用户名、密码）发送给RADIUS服务器进行认证；

(5) RADIUS服务器认证该用户，如果认证通过则返回该用户对应的LNS地址等相关信息，并且LAC准备发起Tunnel连接请求；

(6) LAC端向指定LNS发起Tunnel连接请求；

(7) LAC端向指定LNS发送CHAP challenge信息，LNS回送该challenge响应消息CHAP response，并发送LNS侧的CHAP challenge，LAC返回该challenge的响应消息CHAP response；

(8) 隧道验证通过；

(9) LAC端将用户CHAP response、response identifier和PPP协商参数传送给LNS；

(10) LNS将接入请求信息发送给RADIUS服务器进行认证；

(11) RADIUS服务器认证该请求信息，如果认证通过则返回响应信息；

(12) 若用户在LNS侧配置强制本端CHAP认证，则LNS对用户进行认证，发送CHAP challenge，用户侧回应CHAP response；

(13) LNS再次将接入请求信息发送给RADIUS服务器进行认证；

(14) RADIUS服务器认证该请求信息，如果认证通过则返回响应信息；

(15) 验证通过，用户访问企业内部资源。

注：此节如下附加内容涉及到报文协商，前面没有介绍，可以先了解L2TP笔记2中关于报文格式与协商的相关内容，再回头深入分析此节，此节放在这里是为了便于对于整体协商过程进行深入分析。

### 附加:2、针对流程中步骤7、步骤12的Challenge验证过程的分析

(1)LAC向LNS发SCCRQ请求消息时，会产生一个随机的字符串做为本端CHAP Challenge发给LNS。

(2)LNS收到这个Challenge后，再加上本端配置的密码及SCCRP产生一个新的字符串，用MD5算出一个16个字节的Response，在SCCRP消息中发给LAC。

   同时也产生一个随机的字符串(LNS Challenge)放在SCCRP中一起发给LAC。

(3)LAC将自己的CHAP Challenge加上本端配置的密码，再加上SCCRP产生一个新字符串，用MD5算出一个16字节的字符串，并与LNS发来的SCCRP中带的LNS CHAP Response比较，相同通过，不同断掉。

(4)同理LNS也要验证LAC，LAC用在SCCRP中发现的LNS CHAP Challenge加上本端密码和SCCCN组合，再用MD5算出一个16字节的字符串做为LAC CHAP Response在SCCCN中发给LNS。

(5)LNS用自己发的Challenge+本端密码+SCCCN用MD5算出一个16字节字符串，与收到的作比较，相同通过，不同断掉。

**附加:3、LAC代LNS与PC协商LCP(即认证代理)和用于认证的AVPS**

**正常用户认证方式：**

 当LAC检测到有用户拨入电话的时候，向LNS发送ICRQ，请求在已经建立的tunnel中开始session的建立，LAC可以一直等到接收到了LNS回应的ICRP后，表明session可以建立的时候再回答远端(拨号用户)的呼叫，这样LNS可获得足够的信息来决定是否回答这个远端的呼叫。

**LAC代LNS与PC协商LCP(即认证代理)：**

 LAC在接收到ICRP之前，自行先回答远端(拨号用户)的呼叫，代替LNS与其进行LCP协商和PPP认证，用获得的信息来决定选择哪个LNS(此处对应流程图中步骤5)，这种情况下LAC对呼叫的指示和呼叫的回答只是哄骗性质的。在session可以建立时，LAC向LNS发送ICCN时会携带着先前与呼叫用户进行的LCP协商和PPP认证涉及的特性信息(此处对应流程图中步骤9)，包含这些信息的AVP如下：

**①Inital Received LCP CONFREQ(ICCN\属性26)：**为LNS提供LAC从PPP对端(即拨号用户)接收到初始的conf-request。

**②Last Sent LCP CONFREQ(ICCN\属性27)：**提供LAC发送到PPP对端最后的conf-request。

**③Last Received LCP CONFREQ(ICCN\属性28)：**提供LAC从PPP对端接收到的最后的conf-request。

**④Proxy Authen Type(ICCN\属性29)：**标识是否使用验证代理，验证的类型

0---Reserved                       1---Textual username/password exchange

2---PPP CHAP                     3---PPP PAP

4---No Authentication        5---Microsoft CHAP Version 1(MSCHAPv1)

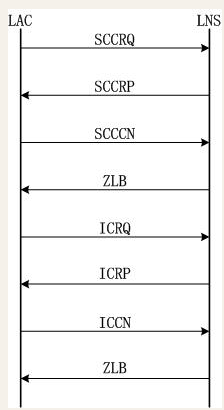
**⑤Proxy Authen Name(ICCN\属性30)：**验证时客户端响应的名称。

**⑥Proxy Authen Challenge(ICCN\属性31)：**LAC发送到PPP对端的Challenge。

**⑦Proxy Authen ID(ICCN\属性32)：**为LAC和PPP对端的验证定了一个ID。

**⑧Proxy Authen Response(ICCN\属性33)：**LAC从PPP对端接收到的PPP Authentication Response。

# 五、L2TP的Tunnel和Session建立过程



**①** 1(SCCRQ)Start-Control-Connection-Request

      控制链接发起请求，由LAC或LNS向对端发送，用来初始化LAC和LNS之间的tunnel，开始tunnel的建立过程。

**②** 2(SCCRP)Strat-Control-Connection-Reply

      表示接受了对端的连接请求，tunnel的建立过程可以继续。

**③** 3(SCCCN)Start-Control-Connection-Connected

       对SCRRP的回应，完成tunnel的建立。

**④** ZLB

      零长度消息报文，一般为查询报文，LAC可以用Hello报文进行恢复，也可以直接丢弃。

**⑤**10(ICRQ)Incoming-Call-Request

       当LAC检测到有用户拨入电话的时候，向LNS发ICRQ，请求在已建立的tunnel中建立session。

**⑥**11(ICRP)Incoming-Call-Reply

      用来回应ICRQ，表示ICRQ成功，LNS也会在ICRP中标识L2TP session必要的参数。

**⑦** 12(ICCN)Incoming-Call-Connected

      用来回应ICRP，L2TP session建立完成。

# 六、其它主要控制消息了解

**控制报文：** 0(reserved)

                       1(SCCRQ)

                       2(SCCRP)

                       3(SCCCN)

                       4(STOPCCN)Stop-Control-Connection-Notification

                       由LAC或LNS发出，通知对端tunnel将要停止，控制链接将要关闭。另外，所有活动的会话都会悄悄的被清除，原因会显示在Result Code AVP中。

                       5(reserved)

                       6(HELLO)

**会话报文：**

                        8(OCRP)Outgoing-Call-Reply

                        9(OCCN)Outgoing-Call-Connected

                        10(ICRQ)

                        11(ICRP)

                        12(ICCN)

                        13(reserved)

                        14(CDN)Call-Disconnect-Notify

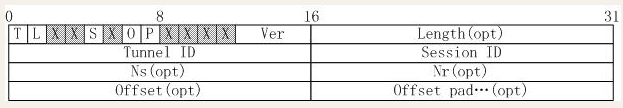
                        由LAC或LNS发出，通知对端session将要停止。

**错误报告报文：**15(WEN)WAN-Error-Notify

**PPP链接控制报文：**16(SLI)Set-Link-Info

7(OCRQ)Outgoing-Call-Request                                       LNS发向LAC用来设置PPP协商的选项，这些选项在call的生存周期中可以随时变化，因此LAC有能力在活动的PPP session上更新内部的呼叫信息和行为。

# 七、L2TP报文头格式



Type(T)：标识消息的类型，0表示是数据消息，1表示控制消息。

Length(L)：置1时，说明Length域的值是存在的，对于控制消息L位必须置1。

X bit：保留位，所有保留位均置0。

Sequence(S)：置1时，说明Ns和Nr是存在的，对于控制消息S必须置1。

Offset(O)：置1时，说明Offset Size域是存在的，对于控制消息O必须置0。

Priority(P)：只用于数据消息，对于控制消息P位置0，当数据消息此位置1时，说明该消息在本列队和传输时应得到优先处理。

Ver：必须是2，表示L2TP数据报头的版本。

Length：标识整个报文的长度(以字节为单位)。

Tunnel ID：标识L2TP控制链接，L2TP Tunnel标识符只有本地意义，一个Tunnel两端被分配的Tunnel ID可能会不同，报头中的Tunnel是指接收方的Tunnel ID，而不是发送方的。本端的Tunnel ID在创建Tunnel时分配。通过Tunnel ID AVPs和对端交换Tunnel ID信息。

Session ID：标识Tunnel中的一个session，只有本地意义，一个session两端Session ID可能不同。

Ns：标识发送数据或控制消息的序号，从0开始，以1递增，到216再从0开始。

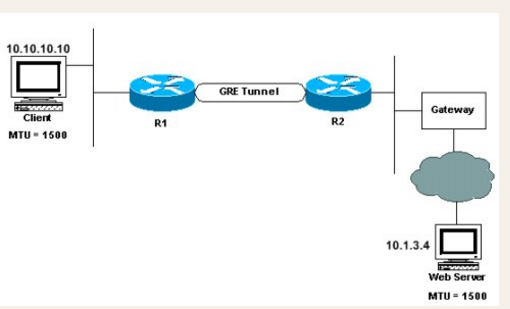
Nr：标识下一个期望接收到的控制消息。Nr的值设置成上一个接收到的控制消息的Ns+1。这样是对上一个接收到的控制消息的确认。数据消息忽略Nr。

Offset Size：如果值存在的话，标识有效载荷数据的偏移。

# Ipsec VPN + GRE隧道的MTU设置详细解析

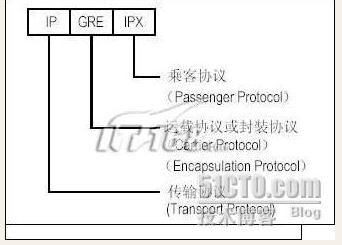
 Cisco给出的Ipsec VPN MTU设置标准是1400字节，MSS值是1360。一般平时都这样设置，但为什么这样设置呢。

以下是GRE Tunnel MTU 设置的几个例子



1. GRE隧道的情况

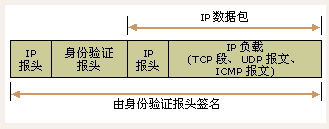
   利用GRE隧道技术封装之后，原来的数据包会在IP报头前面加入一个新的IP报头和新的GRE报头。如下图，假设原始数据报头是IPX,那么在IP GRE的网络中传输就会封装上新IP报头＋GRE报头



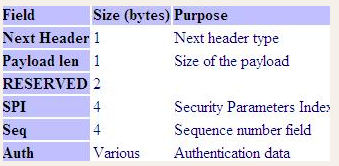
注意：新的IP报头长度20字节，GRE报头最小封装是4字节.那就是说现在的MTU必须调成1476字节，（1500-20-4=1476）。否则将会自动分片，如果你把DF位设置位1，就丢包。

2. Ipsec VPN分2种情况，一种是隧道模式（PS:IPsec的隧道模式与GRE隧道不是一个概念，别混肴！），另一种是传输模式。传输模式只对IP负载部分进行加密,而在传输模式的基础上对IP负载的传输过程进行保护，需要用到隧道模式。Cisco默认的Ipsec VPN就是隧道模式，如果你改为传输模式那数据只加密，而没有在传输过程作保护的，当然传输模式下遇到Man-in-middle的话，Hacker貌似也无法解密数据。(PS:记住啦，隧道模式是需要在已经加密的IP负载基础上加入新的IP报头+AH报头或ESP报头滴!!!)

* 1. 隧道模式中的以IP报头+AH报头封装的数据包格式

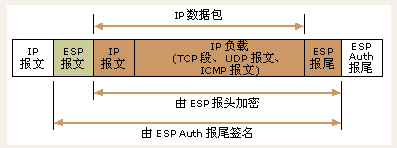


AH报头的格式以及最小长度字节示意图

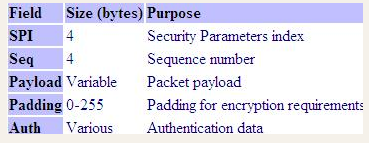


注意：新的IP报头长度20字节，AH报头大小是可变的，但最小长度不能小于12字节（是不能小于12字节，并不是说AH报头就是12字节，至于最大字节限制有待验证!!!）那就是说现在的MTU最多也只能是(1500-20-12=1468)字节才能满足Ipsec VPN 隧道模式的条件。

* 1. 隧道模式中的以IP报头+ESP报头封装的数据包格式



ESP报头的格式以及最小长度字节示意图



注意：新的IP报头长度20字节, ESP报头大小也是可变的，但最小长度不能小于10字节那就是说，现在的MTU最多也只能是(1500-20-10=1470)字节才能满足Ipsec VPN 隧道模式的条件。

3. Ipsec + GRE隧道的的情况

   其实现在已经很容易算出Ipsec+GRE隧道需要占用多小字节了，当用AH封装的时候格式是（新的IP报头+GRE报头+新的IP报头+最小AH报头=最小56字节)，即此时MTU最多只能设置(1500-20-4-20-12=1444字节)才能满足Ipsec + GRE的条件。同理，ESP封装的时候MTU最多只能设置1446字节。

[背景知识]  
MTU: Maxitum Transmission Unit 最大传输单元

MSS: Maxitum Segment Size 最大分段大小（偶是直译，翻译的不好，不要打  
俺PP）

[PPP](http://www.net130.com/CMS/Pub/network/network_protocal/04314.htm)oE: [PPP](http://www.net130.com/CMS/Pub/network/network_protocal/04314.htm) Over Ethernet（在以太网上承载[PPP](http://www.net130.com/CMS/Pub/network/network_protocal/04314.htm)协议）

[分析过程]  
先说说这MTU最大传输单元，这个最大传输单元实际上和链路层协议有着密切的关系，让我们先仔细回忆一下EthernetII帧的结构DMAC+SMAC+Type+Data+CRC  
由于以太网传输电气方面的限制，每个以太网帧都有最小的大小64bytes最大不能超过1518bytes，对于小于或者大于这个限制的以太网帧我们都可以视之为错误的数据帧，一般的以太网转发设备会丢弃这些数据帧。  
（注：小于64Bytes的数据帧一般是由于以太网冲突产生的“碎片”或者线路干扰或者坏的以太网接口产生的，对于大于1518Bytes的数据帧我们一般把它叫做Giant帧，这种一般是由于线路干扰或者坏的以太网口产生）

由于以太网EthernetII最大的数据帧是1518Bytes这样，刨去以太网帧的帧头（DMAC目的MAC地址48bit=6Bytes+SMAC源MAC地址48bit=6Bytes+Type域2bytes）14Bytes和帧尾CRC校验部分4Bytes（这个部门有时候大家也把它叫做FCS），那么剩下承载上层协议的地方也就是Data域最大就只能有1500Bytes这个值我们就把它称之为MTU。这个就是网络层协议非常关心的地方，因为网络层协议比如IP协议会根据这个值来决定是否把上层传下来的数据进行分片。就好比一个盒子没法装下一大块面包，我们需要把面包切成片，装在多个盒子里面一样的道理。

当两台远程PC互联的时候，它们的数据需要穿过很多的路由器和各种各样的网络媒介才能到达对端，网络中不同媒介的MTU各不相同，就好比一长段的水管，由不同粗细的水管组成（MTU不同）通过这段水管最大水量就要由中间最细的水管决定。

对于网络层的上层协议而言（我们以TCP/IP协议族为例）它们对水管粗细不在意它们认为这个是网络层的事情。网络层IP协议会检查每个从上层协议下来的数据包的大小，并根据本机MTU的大小决定是否作“分片”处理。分片最大的坏处就是  
降低了传输性能，本来一次可以搞定的事情，分成多次搞定，所以在网络层更高一层（就是传输层）的实现中往往会对此加以注意！有些高层因为某些原因就会要求我这个面包不能切片，我要完整地面包，所以会在IP数据包包头里面加上一  
个标签：DF（Donot Fragment）。这样当这个IP数据包在一大段网络（水管里面）传输的时候，如果遇到MTU小于IP数据包的情况，转发设备就会根据要求丢弃这个数据包。然后返回一个错误信息给发送者。这样往往会造成某些通讯上的问题，不过幸运的是大部分网络链路都是MTU1500或者大于1500。

对于UDP协议而言，这个协议本身是无连接的协议，对数据包的到达顺序以及是否正确到达不甚关心，所以一般UDP应用对分片没有特殊要求。

对于TCP协议而言就不一样了，这个协议是面向连接的协议，对于TCP协议而言它非常在意数据包的到达顺序以及是否传输中有错误发生。所以有些TCP应用对分片有要求---不能分片（DF）。

花开两朵，各表一枝，说完MTU的故事我们该讲讲今天的第二个猪脚---[PPP](http://www.net130.com/CMS/Pub/network/network_protocal/04314.htm)oE所谓[PPP](http://www.net130.com/CMS/Pub/network/network_protocal/04314.htm)oE就是在以太网上面跑[PPP](http://www.net130.com/CMS/Pub/network/network_protocal/04314.htm" \t "_blank)协议，有人奇怪了，[PPP](http://www.net130.com/CMS/Pub/network/network_protocal/04314.htm)协议和Ethernet不都是链路层协议吗？怎么一个链路层跑到另外一个链路层上面去了，难道升级成网络层协议了不成。其实这是个误区：就是某层协议只能承载更上一层协议。

为什么会产生这种奇怪的需求呢？这是因为随着宽带接入（这种宽带接入一般为Cable Modem或者xDSL或者以太网的接入）由于以太网缺乏认证计费机制而传统运营商是通过[PPP](http://www.net130.com/CMS/Pub/network/network_protocal/04314.htm)协议来对拨号等接入服务进行认证计费的，所以就出了这么一个怪胎：[PPP](http://www.net130.com/CMS/Pub/network/network_protocal/04314.htm)oE。（有关[PPP](http://www.net130.com/CMS/Pub/network/network_protocal/04314.htm)oE的详细介绍参见V大以及本站其他成员的一些介绍文章，我就不啰里啰唆的了）

[PPP](http://www.net130.com/CMS/Pub/network/network_protocal/04314.htm)oE带来了好处，也带来了一些坏处，比如：二次封装耗费资源，降低了传输效能等等，这些坏处俺也不多说了，最大的坏处就是[PPP](http://www.net130.com/CMS/Pub/network/network_protocal/04314.htm)oE导致MTU变小了以太网的MTU是1500，再减去[PPP](http://www.net130.com/CMS/Pub/network/network_protocal/04314.htm)的包头包尾的开销（8Bytes），就变成1492。

如果两台主机之间的某段网络使用了[PPP](http://www.net130.com/CMS/Pub/network/network_protocal/04314.htm)oE那么就会导致某些不能分片的应用无法通讯。

这个时候就需要我们调整一下主机的MTU，通过降低主机的MTU，这样我们就能够顺利地进行通讯了。

当然对于TCP应用而言还有另外的解决方案。  
马上请出今天第三位猪脚：MSS。  
MSS最大传输大小的缩写，是TCP协议里面的一个概念。  
MSS就是TCP数据包每次能够传输的最大数据分段。为了达到最佳的传输效能TCP协议在建立连接的时候通常要协商双方的MSS值，这个值TCP协议在实现的时候往往用MTU值代替（需要减去IP数据包包头的大小20Bytes和TCP数据段的包头20Bytes）所以往往MSS为1460。通讯双方会根据双方提供的MSS值得最小值确定为这次连接的最大MSS值。

介绍完这三位猪脚s  
我们回过头来看前言里面的那个问题，我们试想一下，如果我们在中间路由器上把每次TCP连接的最大MSS进行调整这样使得通过[PPP](http://www.net130.com/CMS/Pub/network/network_protocal/04314.htm)oE链路的最大MSS值加上数据包头包尾不会超过[PPP](http://www.net130.com/CMS/Pub/network/network_protocal/04314.htm)oE的MTU大小1492这样就不会造成无法通讯的问题.所以上面的问题可以通过ip tcp adjust-mss 1452来解决。

当然问题也可以通过修改PC机的MTU来解决。

## 网络排障总结

 1章 故障处理方法  
  
　　一、网络的复杂性  
　　一般网络包括路由、拨号、交换、视频、WAN（ISDN、帧中继、ATM、…）、LAN、VLAN、…  
　　二、故障处理模型  
　　1、 界定问题（Define the Problem）  
　　详细而精确地描述故障的症状和潜在的原因  
　　2、 收集详细信息（Gather Facts）R>信息来源：关键用户、网络管理系统、路由器/交换机  
　　1） 识别症状：  
　　2） 重现故障：校验故障依然存在  
　　3） 调查故障频率：  
　　4） 确定故障的范围：有三种方法建立故障范围  
　　? 由外到内故障处理（Outside-In Troubleshooting）：通常适用于有多个主机不能连接到一台服务器或服务器集  
　　? 由内到外故障处理（Inside-Out Troubleshooting）：  
　　? 半分故障处理（Divide-by-Half Troubleshooting）  
　　3、 考虑可能情形（Consider Possibilities）考虑引起故障的可能原因  
　　4、 建立一份行动计划（Create the Action Plan）  
　　5、 部署行动计划（Implement the Action Plan）  
　　用于纠正网络故障原因。从最象故障源处，想出处理方法每完成一个步骤，检查故障是否解决  
　　6、 观察行动计划执行结果（Observe Results）  
　　7、 如有行动计划不能解决问题，重复上述过程（Iterate as Needed）  
  
　　三、记录所做修改  
　　在通过行动计划解决问题后，建议把记录作为故障处理的一部分，记录所有的配置修改。  
第2章 网络文档  
  
　　一、网络基线  
　　解决网络问题的最简单途径是把当前配置和以前的配置相比较。  
　　基线文档由不同的网络和系统文档组成，它包括：  
　　? 网络配置表  
　　? 网络拓扑图  
　　? ES网络配置表  
　　? ES网络拓扑图  
　　创建网络的注意事项：  
　　1） 确定文档覆盖的范围；  
　　2） 保持一致：收集网络中所有设备的相同信息；  
　　3） 明确目标：了解文档的用途；  
　　4） 文档易于使用和访问；  
　　5） 及时维护更新文档。  
  
　　二、网络配置表  
　　网络配置表的通常目标是提供网络中使用的硬件和软件组成的列表，其组成有：  
　　分级 项目  
　　杂项信息设备名、设备型号、CPU类型、FLASH、DRAM、接口描述、用户名口令  
　　第1层 介质类型、速率、双工模式、接口号、连接插座或端口  
　　第2层 MAC地址、STP状态、STP根桥、速端口信息、VLAN、Etherchannel配置、封装、中继状态、接口类型、端口安全、VTP状态、VTP模式  
　　第3层 IP地址、IPX地址、HSRP地址、子网掩码、路由协议、ACL、隧道信息、环路接口  
　　在多数情形下，存储这些信息的最佳方式是电子表格或数据库，电子表格用于较小的网络，数据库用于较大的网络。  
  
　　三、网络拓扑图  
　　网络拓扑图是图示网络的各组成部分之间如何在逻辑上和物理上相互连接。  
　　1、网络拓扑图的组成  
　　分级 项目  
　　杂项信息 设备名、设备型号、设置间连接、接口描述  
　　第1层 介质类型、接口号  
　　第2层 MAC地址、VLAN、封装、中继状态、接口类型、DLCI  
　　第3层 IP地址、子网掩码、路由协议  
　　对于大型的网络，可以制作多个网络拓扑图，每个网络拓扑图反映一个分离的部分。  
　　2、建立网络拓扑图  
  
　　四、发现网络配置信息  
　　1、收集路由器和第3层交换机网络配置信息  
　　show version ；显示设备型号、Flash、DRAM、IOS版本  
　　show ip interface brief ；显示接口简要信息（类型、状态、协议状态、IP地址）  
　　show interface e0/0 ；显示某接口详细信息（MAC、IP、MASK、…）  
　　show ip protocols ；显示IP路由协议信息  
　　show ip interface e0/0 ；显示接口的IP协议信息（状态、IP地址、ACL、…）  
　　2、收集交换机配置信息  
　　交换机网络配置表包含的信息：设备名、型号、位置、Flash、DRAM、CATOS版本、管理地址、VTP域、VTP模式、端口号、端口速率、端口双工、VLAN、STP状态、速端口状态、中继状态、…  
　　show version ；显示IOS或CATOS版本、DRAM、Flash  
　　show vtp domain ；（CatOS）显示VTP域和VTP模式  
　　show vtp status ；（IOS）  
　　show interface ；（CatOS）显示管理接口信息  
　　show port ；（CatOS）显示每个端口的简要信息（号、VLAN、双工、…）  
　　show interface ；（IOS）  
　　show trunk ；（CatOS）显示中继信息（模式、封装、允许端口、剪裁、…）  
　　show interface trunk ；（IOS）  
　　show spantree 45 ；（CatOS）显示端口的STP模式、类型、状态、速端口、…）  
　　show spanning-tree 45 ；（IOS）  
　　3、发现相邻CISCO设备的信息  
　　CDP（Cisco Discovery Protocol）是CISCO的专用协议，用于识别直接相邻的CISCO设备信息，CDP工作在第2层。  
　　Show cdp neighbor ；显示相邻CISCO设备的简要信息（ID、相邻接口、平台、…）  
　　Show cdp neighbor detail；显示相邻CISCO设备的详细信息（包含第3层信息）  
  
　　五、创建网络文档的过程  
　　1、 LOGIN ；登录到设备进入特权模式。  
　　2、 接口发现 ；发现关于设备的所需信息  
　　3、 Document ；在网络配置表中记录发现的信息。  
　　4、 Diagram ；从网络配置表传输所需信息到网络拓扑图  
　　5、 设备发现 ；判断是否有相邻设备没有记录文档。  
第3章 ES文档和故障处理  
  
　　一、ES网络配置表  
　　ES网络配置表是ES的硬件和软件组成的列表。ES网络配置常包括以下项目：  
　　分级 项目  
　　杂项信息系统名、系统厂商/型号、CPU速率、RAM、存储器、系统功能  
　　第1、2层 介质类型、接口速率、VLAN、MAC、网络接头  
　　第3层 IP地址、缺省网关、子网掩码、WINS、DNS、  
　　第7层操作系统（版本）、基于网络的应用程序、高带宽应用程序、低延时应用程序、特定考虑  
  
　　二、ES网络拓扑图  
　　ES网络拓扑图的典型项目有：系统名、网络连接、物理位置、系统目标、VLAN、IP地址、子网掩码、操作系统、网络应用程序  
　　大多数ES网络拓扑图都建立在网络拓扑图中，其中还可加入ES网络配置表数据的子集。  
  
　　三、收集ES网络配置信息  
　　通用命令：  
　　1） ping host/ip-address ；发送和接收ICMP响应，校验网络的连通性  
　　2） arp -a ；查看修改ES的MAC-IP映射表（同一子网）  
　　3） telnet host/ip-address ；登录远程ES或特定TCP端口  
　　Windows平台命令  
　　1） ipconfig /all ；查看修改ES的IP信息（适用所有Windows平台）  
　　2） winipcfg ；查看修改ES的IP信息（仅适用于Win9x平台）  
　　3） tracert host/ip-address ；校验到主机的连接并显示路径上的设备IP  
　　4） route print ；显示本设备IP路由表的内容  
　　5） netstat ；显示当前网络连接  
　　Unix、Linux和Mac OS系统命令  
　　1） ifconfig -a ；查看UNIX和MAC主机的IP信息  
　　2） traceroute host/ip ；  
　　3） route –n ；  
　　4） cat /etc/resolv.conf ；查看DNS服务器信息  
  
　　四、通用的故障处理过程  
　　1、通用的故障处理过程：  
　　l 收集症状：收集网络、用户、ES的症状  
　　1） 分析现存症状  
　　2） 判断所属  
　　3） 窄化范围  
　　4） 判定症状  
　　5） 记录症状  
　　l 分离问题  
　　1） Bottom-Up troubleshooting  
　　从物理层开始向上排查，直到应用层。常用于怀疑问题发生在物理层，或在处理复杂网络问题时使用。  
　　2） Top-Down troubleshooting  
　　从应用层开始向下排查故障，用于怀疑问题发生在软件部分。  
　　3） Divide-and-Conquer troubleshooting  
　　选择OSI模型的特定层（数据链路层、网络层、传输层）开始故障处理，确定问题是在该层、还是上层或下层。适于具有丰富的经验的人员使用。  
　　常用traceroute命令检查下4层（从物理层到应用层）。  
　　l 纠正问题  
　　2、ES故障处理命令  
　　1） ping  
　　连续Ping： ping –t 192.168.0.1 ；Windows系统  
　　ping –s 192.168.0.1 ；Unix环境  
　　记录路由： ping –r 192.168.0.1 ；Windows  
　　ping –s –nRv 192.168.0.1 ；Unix  
　　2） Trace Route  
　　Tracert 10.0.0.1 ；Windows系统  
　　Tracerout 10.0.0.1 ；Unix  
　　Ping记录路由器的出接口，而traceroute通常记录进入的接口。  
　　3） Arp  
　　显示第2层和第3层地址的映射表： Arp –a ；Windows/Unix  
　　4） Route  
　　显示路由表： route print ；windows系统  
　　route –n ；Unix  
　　5） Netstat  
　　显示到ES的当前连接及端口： netstat –n ；Windowx & Unix  
　　6） Ipconfig＆Ifconfig  
　　显示ES的IP配置： ipconfig /all ；windows  
　　ifconfig –a ；unix  
　　7） Nbtstat  
　　显示当前名称解析缓存： nbtstat –c ；  
　　清除当前名称解析缓存： nbtstat –r ；  
第4章 协议属性  
　　一、OSI参考模型  
　　应用层  
　　表示层  
　　会话层  
　　传输层  
　　网络层  
　　数据链路层  
　　物理层  
  
　　二、全局协议分类  
　　1、面向连接的协议：  
　　windows size：在需要目标系统确认的传输的数据包数。  
　　队列数据传送：对进入和发送的PDU指定序号，在目的地再按序号重排数据；  
　　流控：确保发送的速率不超过目标接收的速率，通过为传输建立窗口尺寸实现；  
　　错误控制：确保接收到的数据连续并无错，如有丢失或损失的PDU，则不发送ACK包。  
　　面向连接的协议有：ATM、TCP、Novell SPX、Apple Talk ATP；  
　　2、非连接的协议  
　　不包括连接设置和终止，没有流控和错误控制。  
　　非连接的协议有：UDP、Apple Talk DDP、Novell IPX；  
  
　　三、第2层：数据链路层  
　　1、Ethernet/IEEE802.3  
　　2、Token Ring/IEEE802.5  
　　四、PPP  
　　五、SDLC  
　　六、Frame Relay  
　　七、ISDN  
　　八、第3、4层：IP路由协议  
　　1、IP  
　　2、ICMP  
　　3、TCP  
　　4、UDP  
  
　　第5层 Cisco测试命令和TCP/IP连接故障处理  
　　一、故障处理命令  
  
　　1、show命令：  
　　1） 全局命令：  
　　show version ；显示系统硬件和软件版本、DRAM、Flash  
　　show startup-config ；显示写入NVRAM中的配置内容  
　　show running-config ；显示当前运行的配置内容  
　　show buffers ；详细输出buffer的名称和尺寸  
　　show stacks ；提供路由器进程和处理器利用率信息, 用stack decode  
　　show tech-support ；显示几个show命令的输出  
　　show access-lists ；查看访问列表配置  
　　show memory ；用于测试内存问题  
　　2） 接口相关命令  
　　show queueing [fair|priority|custom]  
　　show queue e0/1 ；查看接口上队列的设置和操作  
　　show interface e0/1 ；Cisco缺省的Ethernet封装方法是ARPA  
　　show ip interface e0/1 ；显示指定接口的TCP/IP配置信息  
　　3） 进程相关命令  
　　show processes cpu ；显示路由器CPU的使用率和当前的进程  
　　show processes memory ；显示路由器当前进程的内存使用情况  
　　4） TCP/IP协议相关命令  
　　Show ip access-list ；显示IP访问列表（1-199）  
　　Show ip arp ；显示路由器的ARP缓存（IP、MAC、封装类型、接口）  
　　Show ip protocols ；显示运行在路由器上的IP路由协议的信息  
　　Show ip route ；显示IP路由表中的信息  
　　Show ip traffic ；显示IP流量统计信息  
  
　　2、debug命令  
　　DEBUG不应在CPU使用率超过50%的路由器上运行。  
　　1） 限制debug输出  
　　在使用DEBUG获得所需数据后，要关闭Debug  
　　使路由器对所有消息都配置使用时间戳：  
　　Router#service timestamps debug datetime msec localtime  
　　Router#service timestamp log datetime msec localtime  
　　缺省，error和debug信息仅发送到console，telnet到路由器上看不到debug和log的信息。想在telnet中看到debug和log信息：  
　　Router#terminal monitor  
　　Router#terminal monitor ；关闭信息输出  
　　Router#undebug all ；关闭debug进程及所有相关信息的输出  
　　可以应用ACL到debug以限定仅输出要求的debug信息。  
　　如仅查看从10.0.1.1到10.1.1.1的ICMP包：  
　　Router(config)#access-list 101 permit icmp host 10.0.1.1 host 10.1.1.1  
　　Router#debug ip packet detail 101  
　　2） 全局debug命令：  
　　3） 接口debug  
　　4） 协议debug  
　　5） IP debug  
　　debug ip packets  
  
　　3、logging命令  
　　输出error和其它信息到console、terminal、路由器内部buffer或一台syslog服务器：  
　　Router>show logging  
　　Cisco路由器有8种可能的logging级：0-7  
　　Logging级别 名称 描述  
　　1 Emergencies 系统不能用的信息  
　　2 Alerts 直接行动  
　　3 Critical 紧急情形  
　　4 Errors 错误信息  
　　5 Warnings 警告信息  
　　6 Notifications 正常但重要的情形  
　　7 Informational 信息  
　　8 Debugging 调试  
　　缺省地，console、monitor、buffer的logging被设置为debugging级，而trap（syslog）服务器的logging被设置为informational。  
  
  
  
  
4、执行路由核心复制  
　　core dump包含一份当前系统内存中信息的精确拷贝。捕捉包含在内存中信息的方法有：  
　　1） 配置路由器在崩溃时执行Core Dump，存储到TFTP、FTP、RCP服务器：  
　　对TFTP协议，只需指定TFTP服务器IP，不需要任何附加的配置：  
　　Router(config)#exception dump 192.168.1.1 ；TFTP服务器的IP地址  
　　对FTP协议的配置：  
　　Router(config)#exception dump 192.168.1.1 ；FTP服务器的IP地址  
　　Router(config)#ip ftp username Kevin  
　　Router(config)#ip ftp password aloha  
　　Router(config)#ip ftp source-interface e0  
　　Router(config)#exception protocol ftp  
　　对RCP协议的配置：  
　　Router(config)#exception protocol rcp  
　　Router(config)#exception dump 192.168.1.1 ；RCP服务器的IP地址  
　　Router(config)#ip rcmd remote-username Kevin  
　　Router(config)#ip rcmd rcp-enable  
　　Router(config)#ip rcmd rsh-enable  
　　Router(config)#ip rcmd remote-host Kevin 192.168.1.1 kevin ；  
　　2） 在系统没有崩溃的情况下，执行Core Dump命令。  
　　Router#write core  
　　Core Dump仅在Cisco工程师测试和解决路由器问题时有用。  
  
　　5、ping命令  
　　ping用于测试整个网络可达性和连通性。可在用户EXEC模式和特权EXEC模式下使用。  
　　IP的ping使用ICMP协议提供连通性和可能性信息，缺省只发送5个echo信息。  
　　扩展Ping的选项有：源IP地址；服务类型；数据；包头选项。  
　　Ping的响应字符集  
　　字符 解释 字符 解释  
　　! Received an echo-reply message Q Source quench  
　　. Timeout M Unable to fragment  
　　U/H Destination unreachable A Administratively denied  
　　N Network unreachable ? Unknown packet-type  
　　P Protocol unreachable  
　　6、traceroute命令  
　　traceroute用于显示到达目标的包路径。可在用户模式和特权模式下使用。  
　　Traceroute的响应：  
　　字符 解释 字符 解释  
　　Xx msec The RTT for each packet \* Timeout  
　　H Host unreachable U Port unreachable  
　　N Network unreachable P Protocol unreachable  
　　A Administratively denied Q Source quench  
　　? Unknown packet type  
二、LAN连接问题  
　　1、获得IP地址  
　　主机可以动态或静态获得IP地址。  
　　1） DHCP：DHCP比BootP多了地址池和租期。  
　　2） BootP：  
　　3） Helper Addresses：指定集中放置的DHCP服务器的IP地址  
　　Ip helperaddress ip-address ；  
　　No ip forward-protocol udp 137 ；  
　　4） 路由器上的DHCP服务：配置路由器为一台DHCP服务器  
　　5） DHCP和BootP故障处理  
　　Show dhcp server ；  
　　Show dhcp lease ；  
　　2、ARP  
　　ARP映射第2层MAC地址到第3层地址。  
　　Show arp ；显示路由器的ARP表  
　　Debug arp ；  
　　1） ARP代理：缺省Cisco路由器的ARP代理是启用的  
　　在下列情况下，CISCO路由器将用自身的MAC地址响应ARP请求：  
　　? 接收到ARP的接口上的Proxy ARP是启用的；  
　　? ARP请求的地址不在本地子网；  
　　? 路由器的路由表中包含ARP请求地址的子网；  
　　3、TCP连接示例  
  
　　三、IP访问列表  
　　1、标准ACL：基于IP包的源IP地址允许或禁用  
　　2、扩展ACL：提供源地址、目标地址、端口号、会话层协议进行过滤。  
　　3、命名ACL：可以是标准ACL，也可以是扩展ACL。  
　　命名ACL与编号ACL的区别：命名ACL有一个逻辑名，可以删除命名ACL中单独一行。  
　　Ip access-list extended Example-Named-ACL  
　　Deny tcp any any eq echo  
　　Deny tcp any any eq 37  
　　Permit udp host 172.16.10.2 any eq snmp  
　　Permit tcp any any  
第6章 TCP/IP路由协议故障处理  
  
　　一、缺省网关  
　　当包的目的地址不在路由器的路由表中，如路由器配置了缺省网关，则转发到缺省网关，否则就丢弃。  
　　Show ip route ；查看Cisco路由器的缺省网关  
  
　　二、静态和动态路由  
  
　　三、处理k\_protocal/04937.htm" target="\_blank">RIP故障  
　　RIP是距离矢量路由协议，度量值是跳数。RIP最大跳数为15，如果到目标的跳数超过15，则为不可达。  
　　RIP V1是有类别路由协议，RIP V2是非分类路由协议，支持CIDR、路由归纳、VLSM，使用多播（224.0.0.9）发送路由更新。  
　　RIP相关的show命令：  
　　Show ip route rip ；仅显示RIP路由表  
　　Show ip route ；显示所有IP路由表  
　　Show ip interface ；显示IP接口配置  
　　Show running-config  
　　Debug ip rip events ；  
　　常见的RIP故障：RIP版本不一致、RIP使用UDP广播更新  
  
　　四、处理IGRP故障  
　　IGRP是Cisco专用路由协议，距离矢量协议。IGRP的度量值可以基于五个要素：带宽、延时、负载、可靠性、MTU，缺省只使用带宽和延时。  
　　IGRP相关的show命令：  
　　Show ip route igrp ；显示IGRP路由表  
　　Debug ip igrp events ；  
　　Debug ip igrp transactions ；  
　　常见的IGRP故障：访问列表、不正确的配置、到相邻路由器的line down  
  
　　五、处理EIGRP故障  
　　EIGRP是链路状态协议和距离矢量混合协议，是CISCO专用路由协议。EIGRP使用多播地址224.0.0.10发送路由更新，使用DUAL算法计算路由。EIGRP的度量值可以基于带宽、延时、负载、可靠性、MTU，缺省仅使用带宽和延时。  
　　EIGRP使用3种数据库：路由数据库、拓扑数据库、相邻路由器数据库。  
　　EIGRP相关的show命令：  
　　Show running-config  
　　Show ip route  
　　Show ip route eigrp ；仅显示EIGRP路由  
　　Show ip eigrp interface ；显示该接口的对等体信息  
　　Show ip eigrp neighbors ；显示所有的EIGRP邻居及其信息  
　　Show ip eigrp topology ；显示EIGRP拓扑结构表的内容  
　　Show ip eigrp traffic ；显示EIGRP路由统计的归纳  
　　Show ip eigrp events ；显示最近的EIGRP协议事件记录  
　　EIGRP相关的debug命令：  
　　Debug ip eigrp as号  
　　Debug ip eigrp neighbor  
　　Debug ip eigrp notifications  
　　Debug ip eigrp summary  
　　Debug ip eigrp  
　　常见的EIGRP故障：相邻关系、缺省网关等的丢失、老版本IOS的路由、stuck in active。  
　　处理EIGRP故障时，先用show ip eigrp neighbors查看所有相邻路由器，然后再用show ip route gigrp查看路由器的路由表，再用show ip eigrp topology查看路由器的拓扑结构表，也可用show ip eigrp traffic查看路由更新是否被发送。  
六、处理OSPF故障  
　　OSPF是链路状态协议，维护3个数据库：相邻数据库、拓扑结构数据库、路由表。  
　　OSPF相关的show命令：  
　　Show running-config  
　　Show ip route  
　　Show ip route ospf ；仅显示OSPF路由  
　　Show ip ospf process-id ；显示与特定进程ID相关的信息  
　　Show ip ospf ；显示OSPF相关信息  
　　Show ip ospf border-routers ；显示边界路由器  
　　Show ip ospf database ；显示OSPF的归纳数据库  
　　Show ip ospf interface ；显示指定接口上的OSPF信息  
　　Show ip ospf neighbor ；显示OSPF相邻信息  
　　Show ip ospf request-list ；显示链路状态请求列表  
　　Show ip ospf summary-address ；显示归纳路由的再发布信息  
　　Show ip ospf virtual-links ；显示虚拟链路信息  
　　Show ip interface ；显示接口的IP设置  
　　OSPF相关的debug命令：  
　　Debug ip ospf adj ；  
　　Debug ip ospf events  
　　Debug ip ospf flood  
　　Debug ip ospf lsa-generation  
　　Debug ip ospf packet  
　　Debug ip ospf retransmission  
　　Debug ip ospf spf  
　　Debug ip ospf tree  
　　常见的OSPF故障：OSPF的每个area不超过100台路由器，整个网络不超过700台路由器；通配符掩码配置不当；  
  
　　七、处理BGP故障  
　　BGP（包括IBGP和EBGP）的关键配置是邻居关系，BGP使用TCP建立相邻关系。  
　　BGP相关的show命令：  
　　Show ip bgp ；显示BGP所学习到的路由  
　　Show ip bgp network ；显示特定网络的BGP信息  
　　Show ip neighbors ；显示BGP邻居信息  
　　Show ip bgp peer-group ；显示BGP对待组信息  
　　Show ip bgp summary ；显示所有BGP连接的归纳  
　　Show ip route bgp ；显示BGP路由表  
　　BGP相关的debug命令：  
　　Debug ip bgp 192.1.1.1 updates  
　　Debug ip bgp dampening  
　　Debug ip bgp events  
　　Debug ip bgp keepalives  
　　Debug ip bgp updates

九、TCP/IP症状和原因  
　　症状 原因  
　　本地主机不能与远程主机通讯 1） DNS工作不正常2）没有到远程主机的路由3） 缺少缺省网关4） 管理拒绝（ACL）  
　　某个应用程序不能正常工作 1） 管理拒绝（ACL）2）网络没有正常配置以处理该应用程序  
　　启动失败 1） BootP服务器没有MAC地址的实体2） 缺少IP helper-address3） ACL4） 修改NIC或MAC地址5） 重复的IP地址6）不正常的IP配置  
　　不能ping远程主机 1） ACL2） 没有到远程主机的路由3）没有设置缺省网关4） 远程主机down  
　　缺少路由 1） 没有正确配置路由协议2） 发布列表3） 被动接口4）没有通告路由的邻居5） 路由协议版本不一致6） 邻居关系没有建立  
　　相邻关系没有建立 1） 不正确的路由协议配置2） 不正确的IP配置3）没有配置network或neighbor语句4） hello间隔不一致5） 不一致的area ID  
　　高的CPU利用率 1） 不稳定的路由更新2） 没有关闭debug3）进程过重  
　　路由触发活跃模式 1） 不一致的间隔2） 硬件问题3）不稳定的链路  
十、TCP/IP症状和行动计划  
　　问题 行动计划  
　　DNS工作不正常 1）配置DNS主机的配置和DNS服务器，可以使用nslookup校验DNS服务器的工作  
　　没有到远程主机的路由 1） 用ipconfig /all检查缺省网关2） 用show ip route查看是否相应路由3） 如果没有该路由，用show ip route查看是否有缺省网关4）如有网关，检查到目标的下一跳；如无网关，修正问题  
　　ACL 有分离的问题与ACL相关，必须分析ACL、或重写ACL并应用。  
　　网络没有配置以处理应用程序 查看路由器配置  
　　Booting失败 1）查看DHCP或BootP服务器，并查看是否存在故障机的MAC实体2） 使用debug ip udp校验从主机接收的包3） 校验helper-address正确配置4）查看ACL是否禁用包  
　　缺少路由 1） 在第1台路由器上用show ip route查看所学到的路由2）校验相邻路由器3）有正确的路由network和neighbor语句4）对OSPF，校验通配符掩码5） 检查应用到接口上的distribute list6）验证邻居的IP配置7） 如果路由被再发布，验证度量值8）验证路由被正常的再发布  
　　没有构成相邻关系 1） 用show ip protocol neighbors列表已构成的相邻关系2）查看没有构成相邻关系的协议配置3）检查路由配置中的network语句4）用show ip protocol/interface查看特定的接口信息，如Hello间隔  
第7章 处理串行线路和帧中继连接故障  
　　一、处理串行线路故障  
　　1、HDLC封装  
　　High-level Data Link Control（HDLC）是用于串行链路的一种封装方法，HDLC是Cisco路由器串行接口的缺省封装方法。  
　　处理串行链路故障的第一步就是查看链路两端要使用相同的封装类型。  
　　Show interface serial 1 ；查看接口信息  
　　Clear counters serial number ；复位接口的计数器到0  
　　正常情况下，接口和line都是up的。  
　　线缆故障、载波故障和硬件故障都可导致接口down，通过校验电缆连接、更换硬件（包括电缆）、检查载波信令定位问题。  
　　接口up，line down：CSU/DSU故障、路由器接口问题、CSU/DSU或载波的时间不一致、没有从远端路由器接收到keepalive信令、载波问题。应验证本地接口和远端接口的配置。  
　　接口重启的原因：  
　　? 数秒内排队的包没有被发送；  
　　? 硬件问题（路由器接口、线缆、CSU/DSU）；  
　　? 时钟信令不一致  
　　? 环路接口  
　　? 接口关闭  
　　? 线协议down且接口定期重启  
　　show controllers serial 0 ；显示接口状态、是否连有线缆、时钟速率  
　　show buffers ；查看系统buffer池，接口buffer设置  
　　debug serial interface ；显示HDLC或Frame Relay通信信息  
　　2、CSU/DSU环路测试  
　　有四种类型的环路测试：  
　　? 在本地CSU/DSU上测试本地环路；  
　　? 在远端CSU/DSU上测试本地环路；  
　　? 从本地NIU到远端CSU/DSU测试远端环路；  
　　? 从远端NIU到本地CSU/DSU测试远端环路；  
　　用PPP封装的串行链路上，PPP用协商Magic Number检测环回网络。  
　　3、串行线中总结：  
　　1） 症状和问题：  
　　症状或情形 问题  
　　Interface is administratively down;line protocol is down 1）接口被从命令行关闭2）不允许重复的IP地址，两个使用相同IP地址的接口将down  
　　Interface is down;line protocol is down 1） 不合格的线缆2）没有本地提供商的信令3） 硬件故障（接口或CSU/DSU、线缆）4）时钟  
　　Interface is up;line protocol is down 1）未配置的接口：本地或远程2） 本地提供商问题3） Keepalive序号没有增加4） 硬件故障（本地或远端接口、CSU/DSU）5）线路杂音6） 时钟不一致7） 第2层（如LMI）  
　　Interface is up;line protocol is up(looped) 链路在某处环路  
　　Incrementing carrier transition counter 1）来自本地提供商的信号不稳定2） 线缆故障3） 硬件故障  
　　Incrementing interface resets 1） 线缆故障，导致CD信号丢失2）硬件故障3） 线路拥塞  
　　Input drops,errors,CRC,and framing errors 1）线路速率超过接口能力2） 本地提供商问题3） 线路杂音4） 线缆故障5）不合格线缆6） 硬件故障  
　　Output drops 接口传输能力超过线路速率  
　　2） 问题和行动  
　　问题 解决行动方案  
　　本地提供商问题 1）检查CSU/DSU的CD信号和其它信号，看链路是否在发送和接收信息2）如果没有CD信号或有其它问题，联系本地提供商处理故障  
　　不合格或故障的线缆 1） 使用符合设备要求的线缆2）使用breakout盒检查3） 交换故障线缆  
　　未配置的接口 1） 使用show running-config校验接口配置2）确认链路两端使用相同的封装类型  
　　Keepalive问题 1） 验证keepalive被发送2）配置了keepalive发送，debug keepalive3） 验证序号在增加4）如果序号不增加，运行环路测试5） CSU/DSU环路，序号仍不增，则硬件故障  
　　硬件故障 1）更换硬件  
　　接口在环路模式 1） 检查接口配置2）如果在接口配置有环路，移除3）如果接口配置被清除，清除CSU/DSU环路模式4）如CSU/DSU不在环路模式，可能是提供商置环  
　　接口administratively down 1） 检查是否有重复的IP地址2）进行接口配置模式，执行no shutdown  
　　线路速率大于接口能力 1） 使用hold-queue减少进入的队列尺寸2）增加输出的队列尺寸  
　　接口速率大于线路速率 1） 减少广播流量2） 增加输出的队列3）如有需要，使用队列算法  
二、处理帧中继故障  
　　DLCI用于在帧中继中标识虚拟链路，DLCI仅仅是本地信令，DLCI与第3层IP地址相映射。  
　　处理帧中继的步骤：  
　　1） 检查物理层，线缆或接口问题；  
　　2） 检查接口封装；  
　　3） 检查LMI类型；  
　　4） 校验DLCI到IP的映射；  
　　5） 校验Frame Delay的PVC；  
　　6） 校验Frame Delay的LMI；  
　　7） 校验Frame Delay映射；  
　　8） 校验环路测试；  
　　1、帧中继的show命令  
　　show interface  
　　show frame-relay lmi ；显示LMI相关信息（LMI类型、更新、状态）  
　　show frame-relay pvc ；输出PVC信息、每条DLCI的LMI状态、…）  
　　show frame-relay map ；提供DLCI号信息和所有FR接口的封装  
　　2、帧中继的debug命令  
　　debug frame-relay lmi ；显示LMI交换信息  
　　debug frame-relay events ；显示协议和应用程序使用DLCI的细节  
　　3、帧中继总纳  
　　1） 症状和问题  
　　症状或情形 相关问题  
　　Frame Realy link is down 1） 线缆故障2） 硬件故障3）本地服务商问题4） LMI类型不一致5） Keepalive没有被发送6）封装类型不一致7） DLCI不一致  
　　从Frame Delay网络不能ping远端主机 1） DLCI指定了错误的接口2）封装类型不一致3） ACL问题4） 接口配置错误  
　　2） 问题和行动  
　　问题 解决行动方案  
　　线缆故障 1） 检查线缆并测试接头2） 更换线缆  
　　硬件故障 1） 执行环路测试，以分离硬件2）将线缆连接到路由器的另一同样配置的接口，如OK，则需更换硬件  
　　本地服务提供商问题 1）如环路测试使LMI状态up，但不能连接远端着站点，联系本地载波2）包含载波问题，就好象FR配置错误，如DLCI不一致或封装不一致。  
　　LMI类型不一致 1） 校验路由器的LMI类型与PVC上的每个设备都一致2）如使用公共提供商网络，不能访问LMI，与提供商联系  
　　Keepalive问题 1） 使用show interface查看是否keepalive被禁用，或校验keepalive被正常配置2）如果keepalive设置错误，进入配置模式并在接口上指定keepalive间隔  
　　封装类型 1）校验两端路由器的封装方式相同，如有非Cisco路由器，必须用IETF。用show frame-relay命令显示封装信息2）用encapsulation frame-relay ietf更换封装方式，与可用frame-relay map设置某个PVC的封装。  
　　DLCI不一致 1） 用show running-config和show frame-relay pvc显示指派给某接口的DLCI号2）如DLCI号配置正常，联系供应商校验FR交换机是否了相同的DLCI  
　　ACL问题 1） 使用show ip interface显示应用到接口上的ACL2）分析ACL，如有需要，删除或修改它  
第8章 处理ISDN故障  
　　一、ISDN基本原理  
  
　　二、常见ISDN故障  
　　ISDN问题分成3类：配置不当的路由器、物理线缆和ISDN协议、配置不当的交换机。  
　　1、配置不当的路由器  
　　配置不当由于不同原因：typographical错误、从服务供应商提供的错误信息、本路由器配置不正确  
　　1） SPID（Service Profile Identifiers）:如SPID和LDN配置错误，将有ISDN连接问题。SPID仅用于北美，只有服务供应商要求时才设置。  
　　2） CHAP：CHAP认证在使用PPP封装的接口上使用。两端路由器的CHAP配置一定要相同。在PPP中，用户名和口令是大小写敏感的。  
　　3） Dialer Map实体：Dialer map关联高层地址到相关的电话号码。每种协议需要一条dialer map语句。  
　　4） 访问列表：ACL可用于ISDN连接以阻止某类型流量触发连接。  
　　5） PPP：  
　　2、物理层连接  
　　1) BRI：在现有电话线上提供数字服务。  
　　2) ISDN BRI信道：2B+D(2\*64+16+48=192kbps)；ISDN BRI的物理帧为48bits，链路每秒发送4000帧。  
　　3) 本地环路：客户和CO之间的链路，连接ISDN设备到ISDN交换机。  
　　4) 物理层：参考点（R、S、T、U）；设备（LT/ET、NT1、NT2、TE1、TE2、TA）  
  
　　三、配置不当的电话交换机  
　　在新安装ISDN时，必须考虑服务供应商ISDN交换机配置错误的可能性。  
　　1、第2层故障处理：  
　　ISDN第2层故障处理的目标：q.921协议和PPP。  
　　1） q.921：ISDN的第2层在q.921中定义。Q.921信令在D信道上用LAPD协议传输。处理q.921故障最常用命令是debug isdn q921，问题常与TEI（terminal endpoint identifier）、SAPI（service access point identifier）和SABME（set asynchronous balanced mode extended）有关。  
　　TEI＝127表示广播；TEI＝64-126保留用于动态分配。  
　　SAPI=0表示当前第3层信令；63表示用于TEI值分配的管理SAPI；64为呼叫控制。  
　　2） PPP：PPP使用LCP设置和维护链路；NCP配置和维护网络层协议。  
　　2、第3层故障处理：  
　　ISDN第3层也叫q.931，使用debug isdn q931命令可查看call setup、connect、release、cancel、status、disconnect和、user information。  
　　ISDN第3层连接在本地路由器（TE）和远端ISDN交换机（ET）之间。  
　　ISDN呼叫建立的过程：  
　　1） SETUP:在本地TE和远端ET之间发送信息  
　　2） CALL\_PROC：呼叫处理信令  
　　3） ALERT：  
　　4） CONNECT  
　　5） CONNECT\_ACK：  
　　3、交换机类型：  
　　配置ISDN时，必须用isdn switch-type命令指定本地环路的交换机。  
  
  
  
  
四、ISDN故障处理命令  
　　1、ping：在DDR中，ping命令触发一个呼叫，在第2个B信道up前，路由器已完成了ping。  
　　2、clear interface bri n：重置接口上不同的计数器并中止接口上的连接。  
　　3、show interface bri n：显示关于ISDN BRI D信道的信息  
　　4、show interface bri n 1 2：显示ISDN BRI的B信道信息。  
　　5、show controller bri：显示接口硬件控制器信息和U接口，供Cisco的TAC处理故障。  
　　6、show isdn status：显示ISDN接口状态和各层详细信息。  
　　7、show dialer：显示关于DDR连接的信息，包括拨号、成功的连接、IDLE时间、呼叫数。  
　　8、show ppp multilink：  
  
　　五、调试ISDN  
　　1、debug bri：提供有关BRI B信道的信息，包括带宽信息  
　　2、debug isdn q921：获取关于接口D信道的信息，D信息用于在交换机和本地ISDN设备间传输信令。  
　　3、debug dialer：呼叫连接的原因和连接的状态。  
　　4、debug isdn q931：监视发生在第3层的事件。  
　　Cause ID显示呼叫被拒绝的原因；  
　　CallRef ID发送和返回的信息，用于分析路由器和交换机之间不同呼叫的特定会话。  
　　5、debug ppp negotiation：提供建立PPP会话的实时信息，可察看CHAP和PAP验证  
　　6、debug ppp packet：报告实时PPP包流，包括包的类型和所用的B信道第9章交换以太网故障处理  
　　一、Switch、Bridge、Hub  
　　广播域：由Router控制  
　　冲突域：由Switch或Bridge控制  
　　Switch和Hub比较：  
　　类型 Switch Hub  
　　Unicasts 仅发送到目标 发送到所有端口  
　　Broadcasts 发送同VLAN中的所有端口 发送到所有端口  
　　Aggregate bandwidth 等于每个端口的带宽×端口数等于介质速率  
　　Full/half-duplex 可全双工连接 仅半双工  
　　Support for mixed media:Token Ring,Ethernet,FDDI… 依靠switch，可在不同帧类型和物理介质之间传输 仅支持同一介质  
　　混合介质的支持 依赖于桥配置  
　　处理帧 硬件（ASIC） 软件或  
　　端口数量 从4到超过100 通常16个以下  
　　帧类型转换 依靠桥配置  
  
　　二、Catalyst故障处理工具  
　　1、Catalyst命令行接口：  
　　命令行接口有Native模式和Hybrid模式。本机模式配置第3层和第2层在一起；混合模式在不同CLI下配置第3层和第2层，常为基于set的CLI。  
　　2、混合模式下的CLI：  
　　1） show system：关于switch的高级总结信息，包括供电状态、uptime和管理设置  
　　2） show port：显示指定端口或一个模块上所有端口的信息（VLAN、速率、双工、状态、类型、…）  
　　3） show log：报告重要事件，包括所有模块的重启、trap、供电失败、…。  
　　4） show logging buffer：等同于路由器的show log命令，根据logging级别，报告端口up或down、STP、…  
　　5） show interface：报告管理模块上IP配置和SC0接口上VLAN信息。（sl0、sc0）  
　　6） show cdp：显示相邻CISCO设备信息  
　　7） show config：等同于show running-config命令，显示交换机除MSFC等外所有模块上所有设置，仅显示非默认设置。Show config all显示所有设置。  
　　8） show test：仅显示switch管理模块状态，包括接口卡、供电、内存等。  
　　9） show mac：显示大量计数，包括每端口帧流量、发出和进入的帧的总数量、丢弃、…  
　　10）show vtp domain：  
　　11）show cam：显示与端口相关联的MAC地址  
　　12）重复的MAC地址  
　　13）show spantree：显示每个VLAN的SPT进程状态  
　　14）show version：显示硬件和软件版本号，包括内存、系统UP时间统计等  
　　3、RMON（Remote Monitoring）  
　　RMON基于RMONProbe，从电路（物理介质）上采集数据信息。Router和Switch并不支持所有级别的RMON信息，更多的监控可以用SPAN（Switched Port Analyzer交换端口分析，也叫Port Mirroring端口监控）实现。  
　　4、指示灯：  
　　管理引擎上包含有负载LED，可以提示交换机的当前负载。在启动过程中，LED将闪烁；正常情况下，LED常绿；橙色LED提示有问题；红色LED提示有故障。  
三、用STP控制环路  
　　STP算法在802.1D中定义，用于在多交换机时控制重复路径，避免网络环路。  
　　Cisco使用Port fast和Uplink fast时，要防止产生网络环路。  
  
　　四、VLAN  
　　VLAN有基于端口的静态VLAN和基于MAC的动态VLAN  
　　1、ISL：Cisco专用协议，用于连接两台设备以支持多个VLAN。  
　　? ISL只能在支持ISL的产品上使用  
　　? ISL必须是点对点的  
　　? ISL仅用于100Mb全双工  
　　? ISL要求路由器的IOS和内存升级；  
　　? ISL可以支持Token Ring；  
　　? ISL添加30Bit到原始帧；  
　　? ISL在帧的末尾包含CRC。  
　　2、802.1Q：用于连接非Cisco中继到Cisco设备  
　　。  
　　3、VTP：VTP使用多播通知VTP域中所有其它交换机关于域中VLAN的信息。  
　　? VTP服务器：  
　　? VTP客户机  
　　? 透明VTP：  
  
　　五、线缆问题  
　　物理层标准：  
　　线缆 10Mb 100Mb  
　　3类线距离 100m 不可用  
　　5类距离 100m 100m  
　　多模光纤距离 2000m 2000m  
　　单模光纤距离 高达100km 高达100km  
　　1、线缆问题：  
　　1） 万用表（Multimeters）和电缆测试器（Cable Testers）  
　　万用表（Multimeters）和伏欧表（Volt-ohm）用于验证电缆连通性，只能用于测试铜线或其它基于电信号的电缆，不能用于测试光纤。  
　　电缆测试器（Cable Testers）既可测试电缆也可测试光缆，提供给用户更多的被测试电缆的信息，如：连通性、断路、短路、距离过长、噪音、MAC信息、线路负载、…。  
　　2） 时域反射器（TDRs）和光时域反射器（OTDRs）  
　　TDR是更复杂的电缆测试器，可用于定位电?械奈锢砦侍猓觳庠谑裁次恢枚下贰⒍搪贰⒕砬 纫斐Ｏ窒蟆?br />  
　　2、交叉线  
　　交叉线用于两台主机直接相连、连接两台网络设备。  
　　以太网使用1、2、3、6四芯（白橙、橙、白绿、绿），而T1电路使用RJ-45的1、2、3、5四芯  
　　六、交换机连接故障处理  
　　发生在交换机上常见的故障有速率和双工设置，  
　　1、SPAN（交换端口分析器）：  
　　也叫Port Mirroring（端口监视器）交换机拷贝所有被发送到工作站接口的包到另一接口，这个接口没有被指定VLAN。  
　　Set span enable ；配置SPAN  
　　使用SPAN既监视接收的、发送的或所有的包。  
　　2、多层交换特性卡（MSFC）和Catalyst路由：  
　　MSFC是一个在子板的Cisco路由器，安装在管理模块上，提供VLAN间路由。  
　　在CLI下访问MSFC：session  
　　3、路由器和交换机间VLAN：  
　　路由器提供VLAN间的通信。  
　　1） 广播管理：  
　　路由器不转发广播，交换机控制广播仅转发到是源端口所VLAN成员的端口。  
　　2）策略控制：交换机没有策略，而路由器提供连接VLAN的安全和策略控制  
　　3） VLAN交换：经过路由器转发一个包到同VLAN的不同接口  
　　4） VLAN传输：使用不同VLAN协议的两VLAN间或VLAN协议传输到非VLAN第2层协议。  
　　5） 路由：在不同VLAN或非VLAN网络间通信  
　　6） 路由器上VLAN故障处理：  
　　show vlans  
　　show arp  
　　show interface  
　　show cdp neighbor  
　　debug vlan packet  
　　debug spantree  
　　7） show vlans：在路由器上执行，显示路由器VLAN配置的细节，包括：VLAN名、接口、IP地址、VLAN封装协议、接口协议。  
　　8） debug vlan packet：判定在中继上发送到路由器的数据的VLAN。  
　　3、VLAN设计和故障处理  
　　VLAN设计时注意事项：  
　　1） 网络直径要少于8台交换机；  
　　2） VLAN必须在某个限制内进行编号；  
　　七、混合/本地模式命令转换  
　　混合模式 本机模式 解释  
　　Clear vlan No vlan 从配置中删除VLAN  
　　Set cam agingtime Mac-address-table aging-time 设置保留MAC地址的超时值  
　　Set port dulex Duplex 在特定端口上配置双工  
　　Set port name Description 设置端口名  
　　Set port speed speed 设置端口速率  
　　Se tspan Monitor session 设置SPAN端口  
　　Set spantree Spanning-tree 设置STP信息  
　　Set vlan Switchport access vlan 分配某端口到给定VLAN  
　　Show cam dynamic Show mac-address-table dynamic 显示MAC到端口关系  
　　Show port Show interface 显示端口信息  
　　Show span Show monitor 显示SPAN端口  
　　Show test Show diagnostic 显示启动测试结果  
　　Show version Show version 显示交换机IOS版本信息  
　　Show vlan Show vlan 显示VLAN信息  
　　Show vtp domain Show vtp status 显示VTP信息  
第10章 分离并纠正物理层和数据链路层故障  
　　1、识别物理层问题的症状  
　　物理层组件包括：接口／端口、模块、线缆、中继器、网卡、转换器等。  
　　物理层问题将导致链路上数据完全或间断的丢失，应用程序失败，数据传输速率低。  
　　设备的端口和特定部件的LED在正常工作时稳定，故障时LED状态将关闭、闪烁或其它颜色。  
　　物理层问题的常见症状：  
　　2、识别数据链路层问题的症状  
　　数据链路层问题包括：不正常的帧类型（不相符的封装）、重复的MAC地址、换换鹊?层设备的不当行为。  
　　第2层和第3层测试工具（CDP、PING）可以帮助检验并校验数据链路层问题。  
　　3、用于分离物理层和数据链路层问题的命令和应用程序：  
　　1） ES命令：  
　　Ping host|ip-address ；  
　　Arp –a ；  
　　Netstat –rn ；  
　　Ipconfig /all ；  
　　Tracert ；  
　　Winipcfg ；  
　　Ifconfig –a ；  
　　Traceroute ；  
　　2） Cisco IOS命令  
　　Ping ；  
　　Traceroute ；  
　　Debug ；  
　　Show version ；  
　　Show ip interface brief ；  
　　Show interface e 1 ；  
　　Show cdp neighbor detail ；  
　　Show controllers ；  
　　Debug ppp|isdn|serial|asynch|frame-relay  
　　Show arp ；  
　　Debug arp|lapb|stun ；  
　　4、纠正发生在物理层和数据链路层的命令和应用程序  
　　arp –d ；  
　　interface ；  
　　no shutdown ；  
　　encapsulation ；  
　　clock rate ；  
　　controller ；  
　　duplex full|half|auto  
　　speed 10|100|auto  
　　1） 纠正T1|E1问题的命令  
　　channel-group channel-no timeslots timeslot-list speed 56|64  
　　clock source line|internal  
　　framing sf|esf；framing crc4|no-crc4  
　　linecode ami|b8zs；linecode ami|hdb3  
　　pri-group timeslote range  
第11章 分离并纠正网络层问题  
　　1、网络层问题的症状  
　　2、分离网络层问题的ES命令  
　　1） 通用命令：  
　　ping  
　　arp –a  
　　netstat  
　　2） WINDOWS  
　　Route print  
　　Ipconfig /all  
　　Tracert  
　　Winipcfg  
　　3） UNIX&MAC  
　　Ifconfig –a  
　　Traceroute  
　　Route –n  
　　3、分离网络层问题的Cisco IOS命令  
　　1） 通用：  
　　ping  
　　trace  
　　debug  
　　show running-config  
　　2） ARP  
　　Show ip arp  
　　Debug arp  
　　3） 路由表  
　　show ip route  
　　debug ip routing  
　　4） IP接口  
　　Show ip interface brief  
　　5） BGP  
　　Show ip bgp  
　　Show ip bgp summary  
　　Show ip bgp neighbors  
　　Debug ip bgp  
　　6） IP流量  
　　Show ip traffic  
　　Debug ip icmp  
　　Debug ip packet  
　　7） IP访问列表  
　　Show ip access-list

## 链路层常见报文格式及长度

**1. Ethernet帧（除去上层负载后长度为18字节）**

以太帧有好多种，我们最常用到的是Ethernet II

Ethernet II即DIX 2.0：Xerox与DEC、Intel在1982年制定的以太网标准帧格式。Cisco名称为：ARPA



Ethernet II类型以太网帧的最小长度为64字节（6＋6＋2＋46＋4），最大长度为1518字节（6＋6＋2＋1500＋4）。其中前12字节分别标识出发送数据帧的源节点MAC地址和接收数据帧的目标节点MAC地址。（注：ISL封装后可达1548字节，802.1Q封装后可达1522字节）

接下来的2个字节标识出以太网帧所携带的上层数据类型，如下：

IPv4: 0x0800

ARP:0x0806

PPPoE:0x8864

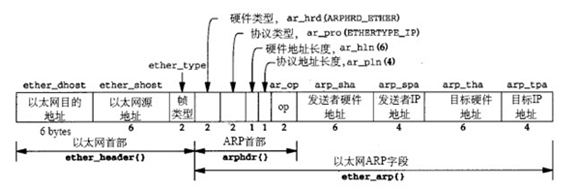
802.1Q tag: 0x8100

IPV6: 0x86DD

MPLS Label:0x8847

在不定长的数据字段后是4个字节的帧校验序列（Frame. Check Sequence，FCS）

**2. ARP （ARP Header长度:8字节）**



硬件类型：1 表示以太网

协议类型：和Ethernet数据帧中类型字段相同

OP操作字段：1 表示ARP请求

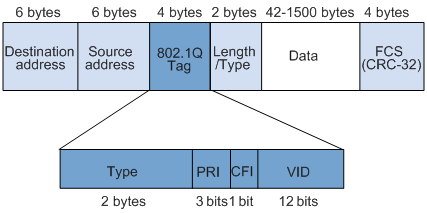
2 表示ARP应答

3 表示RARP请求

4 表示RARP应答

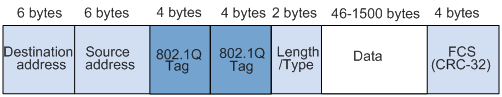
**3. 802.1q VLAN数据帧(4字节）**

**基于802.1Q的VLAN帧格式**



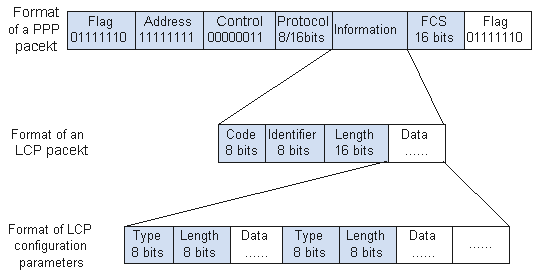
* Type：长度为2字节，取值为0x8100，表示此帧的类型为802.1Q Tag帧。
* PRI：长度为3比特，可取0～7之间的值，表示帧的优先级，值越大优先级越高。该优先级主要为QoS差分服务提供参考依据（COS）。
* VLAN Identifier (VID) : 长度12bits，可配置的VLAN ID取值范围为1～4094。通常vlan 0和vlan 4095预留，vlan1为缺省vlan，一般用于网管。

**QinQ帧格式**



**4. PPP帧（除去信息字段后长度为：8字节）**

PPP报文格式



PPP报文的内容是指Address、Control、Protocol和Information四个域的内容。各字段的含义如下。

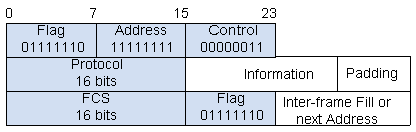
* Flag域Flag域标识了一个物理帧的起始和结束，该字节为0x7E。
* Address域PPP协议是被运用在点对点的链路上，它可以唯一标识对方。因此使用PPP协议互连的两个通信设备无须知道对方的数据链路层地址。所以该字节已无任何意义，按照协议的规定将该字节填充为全1的广播地址。
* Control域同Address域一样，PPP数据帧的Control域也没有实际意义，按照协议的规定通信双方将该字节的内容填充为0x03。Address和Control域一起表示了此报文为PPP报文，即PPP报文头为FF03。
* Protocol域协议域可用来区分PPP数据帧中信息域所承载的数据报文的内容。

|  |  |
| --- | --- |
| **协议代码** | **协议类型** |
| 0021 | Internet Protocol |
| 8021 | Internet Protocol Control Protocol |
| C021 | Link Control Protocol |
| C023 | Password Authentication Protocol |
| C223 | Challenge Handshake Authentication Protocol |

* Information域信息域最大长度是1500字节，其中包括填充域的内容。信息域的最大长度等于PPP协议中MRU（Maximum Receive Unit）的缺省值。

**5. HDLC帧（除去信息字段后长度为：8字节）**

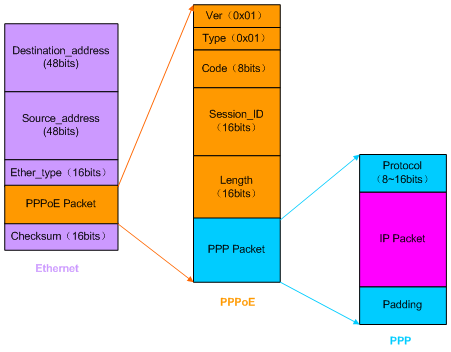
HDLC帧格式



各字段的含义解释：

|  |  |  |
| --- | --- | --- |
| **字段** | **长度（字节）** | **含义** |
| Protocol | 2 | 协议字段。表示Information域中的数据封装的协议类型。 |
| Information | N | 信息字段。可以是任意的二进制比特串，长度未作限定。其上限由FCS字段或通信节点的缓冲容量来决定，目前国际上用得较多的是1000～2000比特，而下限可以是0，即无信息字段。但是监控帧中不可有信息字段。 |

**6. PPPoE报文（报文头长度为6字节）**



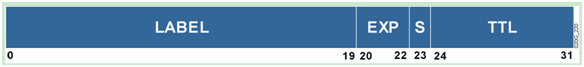
windows系统pppoe MTU大小

默认和最大 PPPoE MTU 大小为 1,480 字节。对于某些 Internet 服务提供商 (ISP)，您可能需要将 PPPoE 连接的 MTU 大小降至 1,400 和 1,480 之间的一个值（例如 1,454）。不要将 MTU 大小设置为小于 1,400。

路由器pppoe拨号时MTU为1492

**7. MPLS Label**

Label报文格式：



MPLS uses a 32-bit label field that contains the following information:

* 20-bit label (a number)
* 3-bit experimental field (usually used to carry IP precedence value)
* 1-bit bottom-of-stack indicator (indicates whether this is the last label before the IP header)
* 8-bit TTL (equal to the TTL in IP header)，used to prevent indefinite looping of packets.

常见网络协议报文格式汇总

1.1       Ethernet数据包格式(RFC894) （除去上层负载后长度为18字节）

Ethernet Ⅱ 帧格式：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 目的MAC地址  6 bytes | 源MAC地址  6 bytes | Type/Length  2 bytes | Payload  46-1500 bytes | FCS(CRC)  4 bytes |

VLAN帧格式：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 目的MAC地址  6 bytes | 源MAC地址  6 bytes |  | Type/Length  2 bytes | Payload | FCS  4 bytes |

|  |  |
| --- | --- |
| 0x8100  2 bytes | TCI 2bytes  3bits:UserPriority   1bit:CFI    12bits:VID |

Ethernet II类型以太网帧的最小长度为64字节（6＋6＋2＋46＋4），最大长度为1518字节（6＋6＋2＋1500＋4）。其中前12字节分别标识出发送数据帧的源节点MAC地址和接收数据帧的目标节点MAC地址。（注：ISL封装后可达1548字节，802.1Q封装后可达1522字节）接下来的2个字节标识出以太网帧所携带的上层数据类型，如下：

IPv4: 0x0800

ARP:0x0806

PPPoE:0x8864

802.1Q tag: 0x8100

IPV6: 0x86DD

MPLS Label:0x8847

基于802.1Q的VLAN帧格式：

Type：长度为2字节，取值为0x8100，表示此帧的类型为802.1Q Tag帧。

PRI：长度为3比特，可取0～7之间的值，表示帧的优先级，值越大优先级越高。该优先级主要为QoS差分服务提供参考依据（COS）。

VLAN Identifier (VID) : 长度12bits，可配置的VLAN ID取值范围为1～4094。通常vlan 0和vlan 4095预留，vlan1为缺省vlan，一般用于网管。

1、DstMac的最高字节的最低BIT位如果为1，表明此包是以太网组播/广播包，送给CPU处理。

2、将DstMac和本端口的MAC进行比较，如果不一致就丢弃。

3、获取以太网类型字段Type/Length。

0x0800→IP               继续进行3层的IP包处理。

0x0806→ARP      送给CPU处理。

0x8035→RARP     送给CPU处理。

0x8863→PPPoE discovery stage  送给CPU处理。

0x8864→PPPoE session stage    继续进行PPP的2层包处理。

0x8100→VLAN

其它值当作未识别包类型而丢弃。

1.2       PPP数据包格式（除去信息字段后长度为：8字节）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 标志7E  1byte | 地址FF  1byte | 控制03  1byte | 协议  2bytes | Payload  最多1500bytes | CRC  2bytes | 标志7E  1byte |

1、获取PPP包类型字段。

0x0021→IP       继续进行3层的IP包处理。

0x8021→IPCP     送给CPU处理。

0xC021→LCP      送给CPU处理。

0xc023→PAP      送给CPU处理。

0xc025→LQR      送给CPU处理。

0xc223→CHAP     送给CPU处理。

0x8023→OSICP    送给CPU处理。

0x0023→OSI      送给CPU处理。

其它值当作未识别包类型而丢弃。

1.3         ARP报文格式(RFC826) **（ARP Header长度:8字节）**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 以太网  目的地址  6bytes | 以太网  源地址  6bytes | 帧类型  2bytes | 硬件类型  2bytes | 协议类型  2bytes | 硬件地址长度  1bytes | 协议地址长度  1bytes | Op  2bytes | 发送端以太网地址  6bytes | 发送端IP地址  4bytes | 目的以太网地址  6bytes | 目的IP地址  4bytes |

|ß----以太网首部---->|ß---------28字节ARP请求/应答------

硬件类型：1 表示以太网

协议类型：和Ethernet数据帧中类型字段相同

OP操作字段：1 表示ARP请求

2 表示ARP应答

3 表示RARP请求

4 表示RARP应答

1.4         IP报文格式(RFC791)(20bytes)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本  4bits | 首部长度  4bits | 服务类型 TOS  8bits | 总长度  16bits | |
| 标识 Ident  16bits | | | 标志 Flag 3bits | 片偏移  13bits |
| TTL  8bits | | 协议号  8bits | 首部检验和  16bits | |
| 源IP地址  32bits | | | | |
| 目的IP地址  32bits | | | | |
| 选项(可选) | | | | |

TOS

1            2              3     4         5        6      7                8

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Precedence | Delay | Throughout | Reliablity | <Reserved> |

1.5         PING报文格式(需IP封装)(8bytes)

|  |  |
| --- | --- |
| IP首部  20bytes | ICMP报文 |

|  |  |  |
| --- | --- | --- |
| 类型(0或8)  8bits | 代码(0)  8bits | 检验和  16bits |
| 标志符 Identi  16bits | | 序号  16bits |
| 选项数据(可选) | | |

1.6         TCP报文格式(需IP封装)(20bytes)

|  |  |  |
| --- | --- | --- |
| IP首部  20bytes | TCP  20bytes | TCP数据 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 源端口号  16bits | | | | | | | | 目的端口号  16bits |
| 序列号  32bits | | | | | | | | |
| 确认序列号  32bits | | | | | | | | |
| 首部长度  4bits | 保留  6bits | U  R  G | A  C  K | P  S  H | R  S  T | S  Y  N | F  I  N | 窗口大小  16bits |
| 检验和  16bits | | | | | | | | 紧急指针  16bits |
|  |  |  |  |  |  |  |  |  |

URG       紧急指针有效

ACK      确认序号有效

PSH      接收方应该尽快将这个报文交给应用层

RST      重建连接

SYN      同步序号用来发起一个连接

FIN      发端完成发送认务

1.7         UDP报文格式(需IP封装)(8bytes)

|  |  |  |
| --- | --- | --- |
| IP首部  20bytes | UDP  8bytes | UDP数据 |

|  |  |
| --- | --- |
| 源端口  16bits | 目的端口  16bits |
| UDP长度  16bits | UDP检验和  16bits |

1.8  MPLS报文格式

|  |  |  |
| --- | --- | --- |
| 二层头 | MPLS标签(可以有多个) | IP报文 |

|  |  |  |  |
| --- | --- | --- | --- |
| Label value  20bits | EXP  3bits | 栈底标记  1bit | TTL  8bits |

MPLS报文类型:

以太网中        0x8847(单播)        0x8848(组播)

PPP类型上   0x8281(MPLSCP)

|  |  |
| --- | --- |
| 协议名称 | 协议号 |
| ICMP | 1 |
| TCP | 6 |
| UDP | 17 |
| OSPF | 89 |

1.9  MTU

|  |  |
| --- | --- |
| 网  络 | MTU字节 |
| 超通道 | 65535 |
| 16Mb/s令牌环(IBM) | 17914 |
| 4 Mb/s令牌环(IEEE 802.5) | 4464 |
| FDDI | 4352 |
| 以太网 | 1500 |
| IEEE 802.3/802.2 | 1492 |
| X.25 | 576 |
| 点对点(低时延) | 296 |

1.10TCP与UDP应用

|  |  |
| --- | --- |
| UDP | TCP |
| RIP | FTP(21) |
| 广播、多播 | TELNET(23) |
| TFTP(S:69) | Rlogin |
| BOOTP(S:67 D:68) | SMTP |
| SNMP  (Admin:162 Proxy:161) |  |

2          以太网帧,TCP/IP数据报文详解

|  |
| --- |
| 其实也不是很详细,详细起来要很厚一本书,但是都是针对主要的写的 以太网2的帧格式  前导码：8个字节，这个主要是给信号同步的，给信号的开始传输的第一个bit定位的  目的地址：6个字节，目的的MAC地址  源地址：6个字节，是自己的MAC地址  类型长度：2字节，辨别上层协议的  数据：46-1500字节，就是上层的所有数据  帧校验FCS：4字节，数据发过去的时候，会随机给个值，对端需要验证这个值，如果值不对，就说明这段数据干扰，或某种原因被修改。  数据部分最小46个字节，最大1500    以太帧最小64字节，最大1518字节。  这层主要是把物理层透明的传输bits组合成字节近而组合成帧，在网络中传输。用MAC地址访问介质，错误发现但是不能纠正。  　  　  IP数据包格式  版本Version：4位，表示IP地址的版本 1111=32bits  首部长度Header Length：4位，表示IP数据报头的长度，最小20字节，最大69字节。那么这个bit是怎么表示的呢。0001 = 1X4=4字节　　0101 = 5　　5X4=20字节　　1111 = 15　　15X4=60字节  优先级与服务类型Priority&type of service：8位，该字段用于表示数据的优先级和服务类型，通过数据包中划分一定的优先级，用于实现服务质量的要求。  总长度Total Length：16位，该字段用于指示整个IP报的长度，最长65535个字节，包括报头和数据。  标识符Identification：16位，当IP数据对上层数据进行分段的时候，它会将所有的段分配一组编号，然后将这些编号放入标识符字段，保证分段不会被错误的进行重组，标识符字段用于表示一个数据包所属的消息，以使得接受节点可以重组被分断或分段的消息。  标志Flags：3位，标志和分段一起被用来传递信息，例如，对当前的包不能进行分段，或者当一个包被分段后，用以指示在一系列的包片段中，最后一个片段是否已发出。  段偏移量Fragment Offset：13位，在一个分段组序列中如何将各片段重新连接起来。  TTL Time to live：8位，IP数据包的生命周期，防御一个数据包在网络中无限循环的转发下去。当TTL值为0时，该数据报将被丢弃，数据包没经过一个路由器，TTL值将减去1。  协议号Protocd：8位，该字段表示在IP数据报中封装的是哪一个协议，TCP为6，UDP为17。  首部校验和Header Checksum：16位，它是16位的错误检测字段，目的的主机和网络中的每个网关都要重新计算包头的校验和，就同源机器所做的一样，如果数据没有被改动，两个计算结果应该是一样的。  源IP地址Source IP Address：32位  目的IP地址Destination IP Address：32位  可选项Options：选项字段，根据实际情况可变长，可以和IP一起使用的选项有多个，例如，可以输入创建该数据报的时间等。  　  　  TCP报文格式  源端口号Source IP Address：16位呼叫端的端口号，这个端口是由发送方随机产生的。  目标端口号Destination Port：16位，被叫端的端口。  32位序列号Sequence Number：由于TCP封包必须要带入IP封包中，所以如果TCP资料太大时（大于IP封包的容许程度），就要进行分段，这个序列号就是记录每个封包的序号，可以让接受端重新将TCP的资料组合起来。  32位确认号Acknowledge   Number：为了确认对方确实收到我们的封包，如果收到了，发送ACK确认。  首部长度：4位，为了确认整个TCP包的大小，就必须要这样来说明数据的起始位。  保留Reserved：6位，未使用的保留位，留着扩展用。  Code：6个控制位，URG，ACK，PSH，RST，SYN，FIN。  Urgent：为1时代表该数据包为紧急封包，根据16位的紧急指针来确定紧急数据最后一个字节的位置，这样接收端就可以有限准确快速的获取紧急数据。  Acknowledge：为1时代表这个封包为回应封包。  Push Function：为1时代表要求对方尽快将数据段送达应用层，无须等待缓冲区满了才送。  Reset：为1时代表通知重新建立TCP连接。  Synchronous：为1时代表发送端希望双方建立同步连接，通常带有SYN标志的封包表示主动连接到对方的意思。  Finish：为1时代表传送结束，所以通知对方是否断线，只是发送端还在等待对方的回应。    窗口Windows：16位，主要是用来控制封包流量的，可以告诉对方目前本身有的缓存器容量还可以接收包，当win=0时，代表缓存器已满，所以应该暂停传输，windows单位，byte。  校验和Checksum：16位，当资料传输时，要由发送端进行检测，并将该动作的校验值标注在这个栏位上，接收者收到这个封包后，会再次对封包进行检测，并且对比与发送方的值是否一致，一致就接收，不一致丢弃，并要求对方重新发送这个封包。  紧急指针Urgent Pointer：16位，和URG配合使用，URG=1时才会产生作用，用来说明紧急数据的末尾字节的位置。  可选项：只有首部长度长于20的时候才有效。  数据：应用层数据分段而得到的数据。 |

IP报文体:  
  
 |------------->16<--------------|------------->16<--------------|  
   0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7   
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-  
 | vesion| IHL   | TypeOfService |          TotleLength          |  
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-  
 |        identification         |flags|      FragmentOffset     |  
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  
 |  Time to Live |    Protocol   |         Header Checksum       |  
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  
 |                       Source Address                          |  
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  
 |                    Destination Address                        |  
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  
 |                    Options                    |    Padding    |  
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  
  
说明：  
Version  -- 版本：4bits长度，指明IP地址的协议版本，如果IPV4，值为4  
IHL      -- 报文头长度：4bits长度，指从Version到Destination Address有几个32位的长度，由上可知有5个32位长  
TOS      -- 服务类型：目前大多数网络对此一般不作处理. bit位域如下  
               0 1 2 3 4 5 6 7  
            -+-+-+-+-+-+-+-+-+  
             |PREC |D|T|R|M|0|  
            precedence -- 优先级：3bits长度.对网络没有强制性  
           （111 - Network Control(网络控制)；110 - Internetwork  Control(互联网络控制)；  
             101 - CRITIC/ECP(至关重要)；100-Flash Override(火速覆盖)；011 - Flash(火速)；  
             010 - Immediate(立即)；001 - Priority(主要)；000 - Routine(常规) ）  
            D  -- 延迟(Delay): 置位代表要求网络能提供低的时延  
            T  -- 吞吐量(Troughout): 置位代表要求网络能提供高的吞吐量  
            R  -- 可靠性(Reliability): 置位表示要求网络提供高的可靠性  
            M  -- 代价(Monetary):  
TotleLen -- IP总长度：16bits长，包括IP报文头和IP报文体的bit长度。IP包的最大长度可达2\u0015\u001F16\u001F-1=65535B≈64kB  
Ident    -- 标识符：16bits长，使目的主机判断新来的分段属于哪个分组，所有属于同一分组的分段包含同样的标识值。  
flags    -- 标志位：3bits长。（Bit0：保留； Bit1：DF； Bit2：MF）  
            DF：置位表示通知路由器不要分段，因为目的端不能重组分段。如果无分段路由器无法转发，则丢弃包。  
            MF：标识是否是最后一个分段。  
FramOset -- 分段偏移：13bits长。标明分段在当前数据报的位置。以8字节(64bits)作为计量单位.第一个分段偏移为0  
TToL     -- 生存时间：8bits长。跳数大小，每经过一个网关时，减1.如果减为0，被抛弃，抛弃的设备将产生ICMP报文  
            发回源主机  
Protocol -- 高层协议：8bits长。标识出传输层的地址或协议号，即报文的数据区是那个高级协议。如VINES、TCP、UDP  
CheckSum -- IP报文头校验和：16bits长。头部所有16bits字的和。crc算法  
SrcAddr  -- 源地址：32bits长，如IP地址  
DesAddr  -- 目的地址：32bits长，如IP地址  
  
Option   -- 报文头选项：0-40bits可变长度  
Padding  -- IP数据报文  
  
  
  
IP报文分段方法：  
                       \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
   原始报文：          |IP Header|         IP Packet data      ......   |  
                       -------------------------------------------------  
  
                       \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
   分段后报文：        |IP Header| IP Packet1 |IP Header| IP Packet2 |    ........  |  
                       -------------------------------------------------------------  
分段时，在原始报文的报文数据域中的合适位置插入分包后的IP协议头。

IP报文体:  
  
 |------------->16<--------------|------------->16<--------------|  
   0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7   
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-  
 | vesion| IHL   | TypeOfService |          TotleLength          |  
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-  
 |        identification         |flags|      FragmentOffset     |  
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  
 |  Time to Live |    Protocol   |         Header Checksum       |  
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  
 |                       Source Address                          |  
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  
 |                    Destination Address                        |  
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  
 |                    Options                    |    Padding    |  
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  
  
说明：  
Version  -- 版本：4bits长度，指明IP地址的协议版本，如果IPV4，值为4  
IHL      -- 报文头长度：4bits长度，指从Version到Destination Address有几个32位的长度，由上可知有5个32位长  
TOS      -- 服务类型：目前大多数网络对此一般不作处理. bit位域如下  
               0 1 2 3 4 5 6 7  
            -+-+-+-+-+-+-+-+-+  
             |PREC |D|T|R|M|0|  
            precedence -- 优先级：3bits长度.对网络没有强制性  
           （111 - Network Control(网络控制)；110 - Internetwork  Control(互联网络控制)；  
             101 - CRITIC/ECP(至关重要)；100-Flash Override(火速覆盖)；011 - Flash(火速)；  
             010 - Immediate(立即)；001 - Priority(主要)；000 - Routine(常规) ）  
            D  -- 延迟(Delay): 置位代表要求网络能提供低的时延  
            T  -- 吞吐量(Troughout): 置位代表要求网络能提供高的吞吐量  
            R  -- 可靠性(Reliability): 置位表示要求网络提供高的可靠性  
            M  -- 代价(Monetary):  
TotleLen -- IP总长度：16bits长，包括IP报文头和IP报文体的bit长度。IP包的最大长度可达2\u0015\u001F16\u001F-1=65535B≈64kB  
Ident    -- 标识符：16bits长，使目的主机判断新来的分段属于哪个分组，所有属于同一分组的分段包含同样的标识值。  
flags    -- 标志位：3bits长。（Bit0：保留； Bit1：DF； Bit2：MF）  
            DF：置位表示通知路由器不要分段，因为目的端不能重组分段。如果无分段路由器无法转发，则丢弃包。  
            MF：标识是否是最后一个分段。  
FramOset -- 分段偏移：13bits长。标明分段在当前数据报的位置。以8字节(64bits)作为计量单位.第一个分段偏移为0  
TToL     -- 生存时间：8bits长。跳数大小，每经过一个网关时，减1.如果减为0，被抛弃，抛弃的设备将产生ICMP报文  
            发回源主机  
Protocol -- 高层协议：8bits长。标识出传输层的地址或协议号，即报文的数据区是那个高级协议。如VINES、TCP、UDP  
CheckSum -- IP报文头校验和：16bits长。头部所有16bits字的和。crc算法  
SrcAddr  -- 源地址：32bits长，如IP地址  
DesAddr  -- 目的地址：32bits长，如IP地址  
  
Option   -- 报文头选项：0-40bits可变长度  
Padding  -- IP数据报文  
  
  
  
IP报文分段方法：  
                       \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
   原始报文：          |IP Header|         IP Packet data      ......   |  
                       -------------------------------------------------  
  
                       \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
   分段后报文：        |IP Header| IP Packet1 |IP Header| IP Packet2 |    ........  |  
                       -------------------------------------------------------------  
分段时，在原始报文的报文数据域中的合适位置插入分包后的IP协议头。