<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。

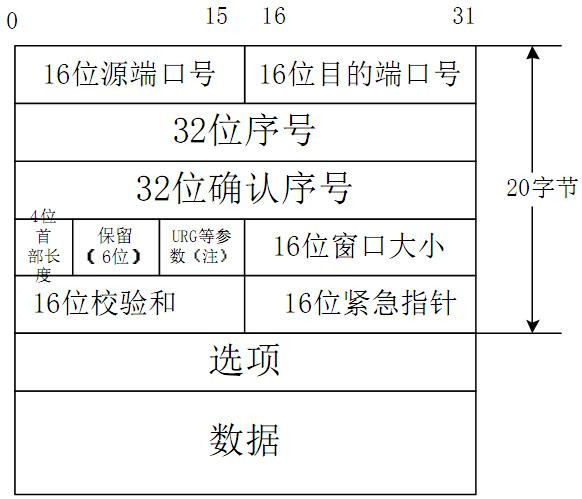
# **IP MTU**

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)：

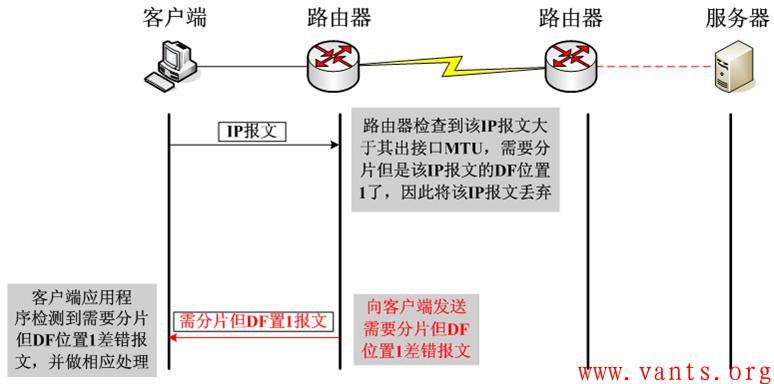
发送符合 RFC 792 中最初定义的“ICMP Destination Unreachable-Fragmentation Needed and DF Set”消息，然后丢弃该包。原始消息格式中不包含有关转发失败的链路的 IP MTU 的信息。（导致PMTU无法正常发现）  
⎫ 发送符合 RFC 1191 中重新定义的“ICMP Destination Unreachable-Fragmentation Needed and DF Set”消息，然后丢弃该包。此新消息格式包含一个 MTU 字段，可指出转发失败的链路的 IP MTU。（PMTU可能会正常发现）

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。

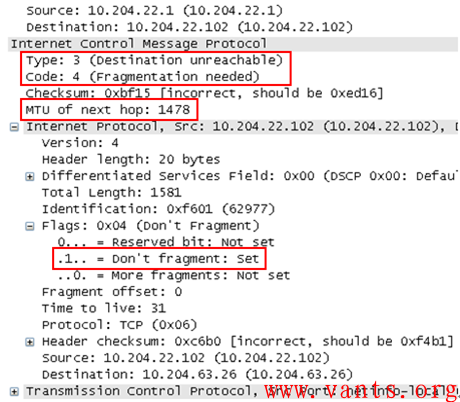
直接丢弃包。直接丢弃需分段但 DF 标记设置为 1 的包的路由器称为 PMTU 黑洞路由器。

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：

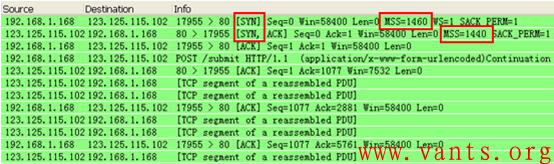


    我们可以看到其差错类型为3，代码为4，并且告知了下一跳的MTU值为1478。在ICMP差错报文里封装导致此差错的原始IP报文的报头（包含IP报头和四层报头）。

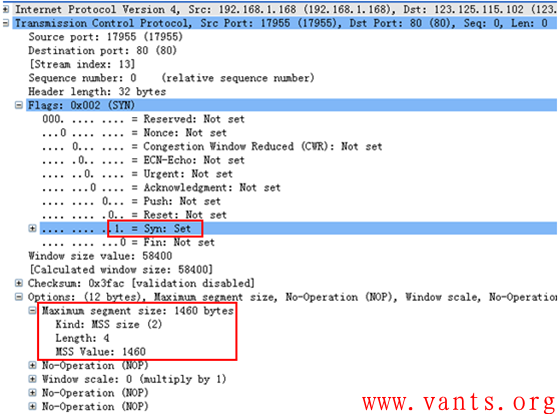
       一旦出现这种因DF位置一而引起丢包，如果客户端无法正常处理的话，将会导致业务应用出现异常，外在表现为页面无法打开、页面打开不全、某些大文件无法传输等等，这将严重影响业务的正常运行。  
那么客户端如何处理这种状况呢？  
TCP主要通过两种方式来应对：  
**1， 协商MSS，在交互之前避免分片的产生  
2， 路径MTU发现（PMTUD）**

**TCP MSS**

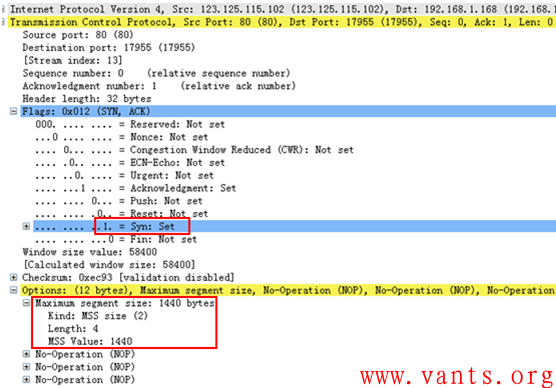
TCP在三次握手建立连接过程中，会在SYN报文中使用MSS（Maximum Segment Size）选项功能，协商交互双方能够接收的最大段长MSS值。  
 MSS是传输层TCP协议范畴内的概念，顾名思义，其标识TCP能够承载的最大的应用数据段长度，因此，MSS=MTU-20字节TCP报头-20字节IP报头，那么在以太网环境下，MSS值一般就是1500-20-20=1460字节。  
客户端与服务器端分别根据自己发包接口的MTU值计算出相应MSS值，并通过SYN报文告知对方，我们还是通过一个实际环境中捕获的数据报文来看一下MSS协商的过程：

[](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：

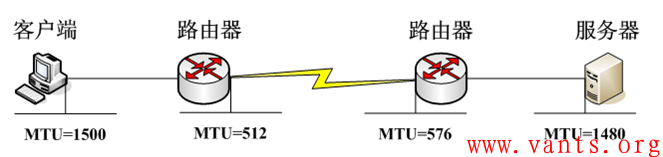
[](http://www.vants.org/content/plugins/kl_album/upload/201209/4150fa5a19d47041aa9727148deca512201209052157112105531868.png)

       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**

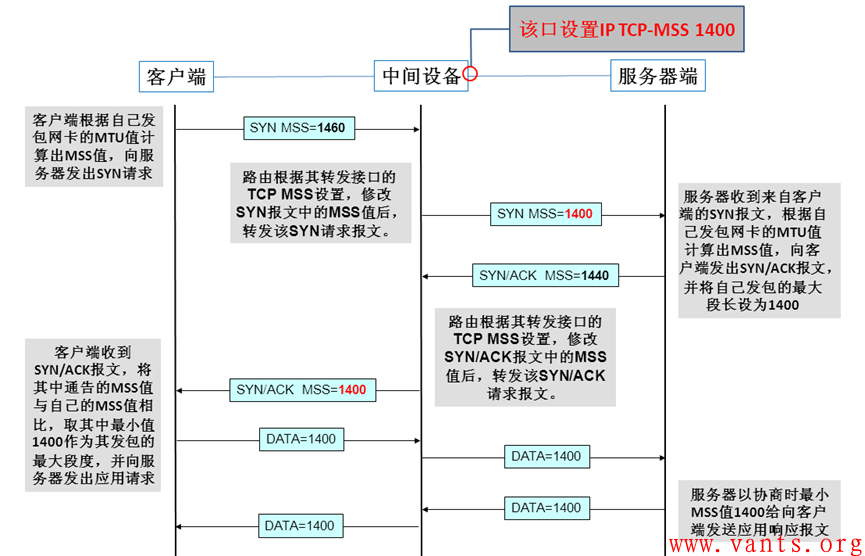
       说起PMTUD，我们必须在此回到上面讲到的ICMP需要分片但DF位置一差错报文，还记得那个ICMP差错报文中有一个字段是告知下一跳的MTU值的吗？PMTUD正是利用ICMP需要分片但DF位置一差错报文的这一特性来实现的。  
       发送方在接收到该差错报文后，会根据该报文给出的下一跳的MTU值计算适合传输的最大段长度，从而在后续的发送报文过程中，避免在中间路径被分片的情况产生。  
       这在端系统主要是通过在路由表里临时添加目的主机路由并将ICMP差错报文告知的下一跳MTU值跟该主机路由关联起来来实现。  
       PMTUD的确是个非常不错的机制，但是在复杂的实际网络环境中，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:

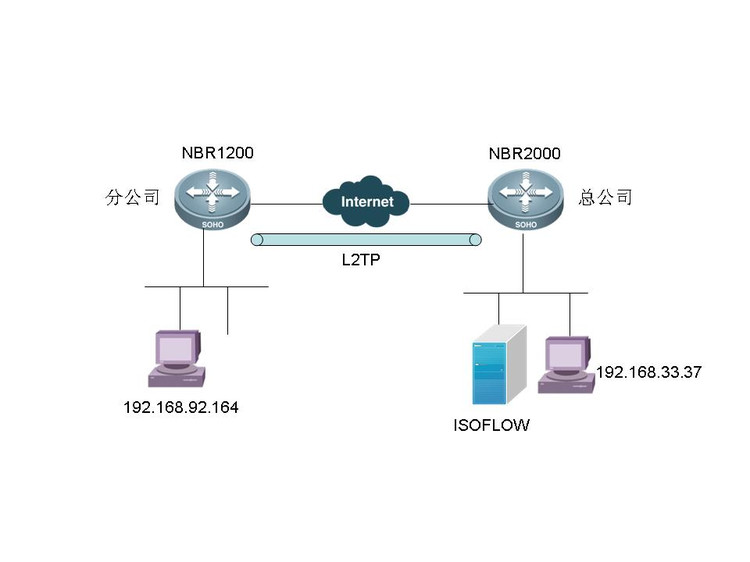


图1-2 某公司网络拓扑示意图

|  |  |  |
| --- | --- | --- |
| 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。`

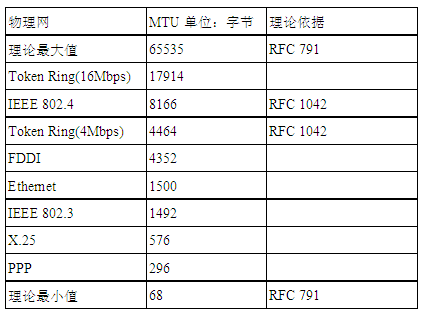
MSS是OSI参考模型中四层的一个概念，即最大分段长度，全称为TCP Maximum Segment Size，指TCP每次能够传输的最大数据分段长度（以字节为单位），MSS一般比MTU小40字节。

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。

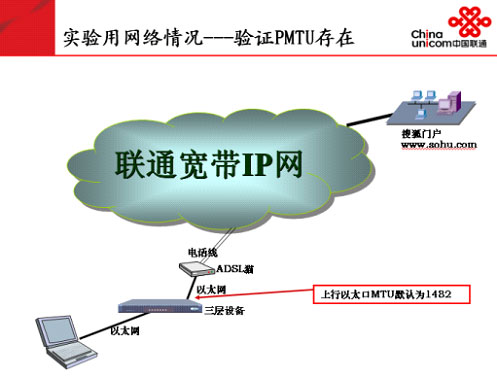
2、PMTU 发现过程：

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)。

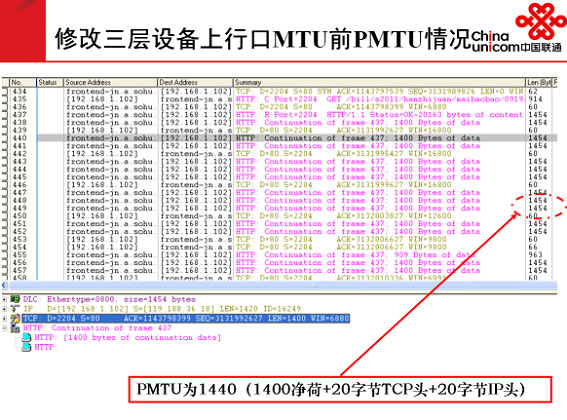
为了达到这个目的，一些操作系统支持自动发现路径MTU的功能，具体过程为：

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]。

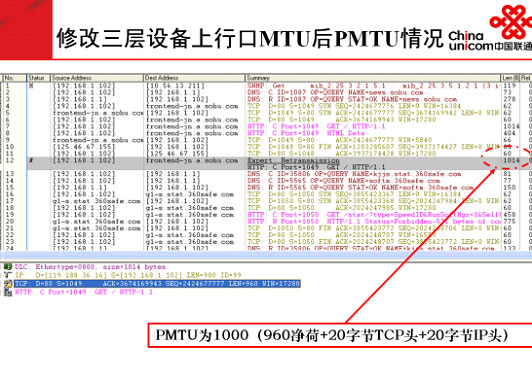
了解了MTU发现的原理，举一个实例验证PMTU变化过程：



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）

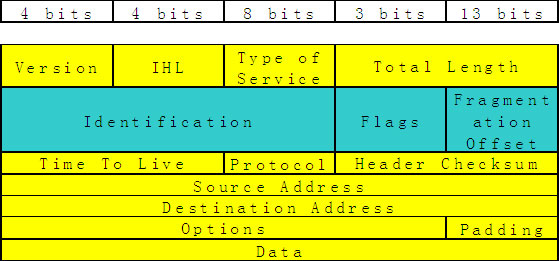
RFC 1191 定义了路径 MTU (PMTU) 发现，它使得源和目的 TCP 对等方能够动态地发现二者之间路径的IP MTU，从而发现该路径的 TCP MSS。一旦收到符合 RFC 1191 定义的“Destination Unreachable-Fragmentation Needed and DF Set”消息，TCP 就会将该连接的 MSS 调整为指定 IP MTU 减去 TCP 和 IP 报头的大小。这样，在该 TCP 连接上发送的后续包就不会超过最大大小，无需分段即可在该路径上传输。

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：

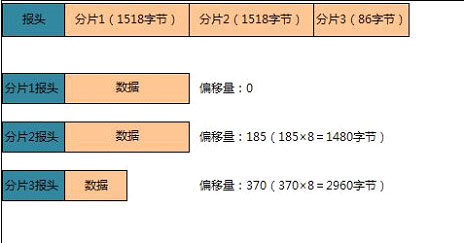
标识符（Identification）：在发送数据包前，发送主机给每个数据包一个ID值，放在16位的标识符字段中。此ID用于标识唯一的数据报或数据流。接收主机利用此ID对收到的数据报进行重组。当分片的IP数据报从源地址发送到目的地址的时候，由于网络延迟或者不同的传输路径的关系，在到达目的主机时，这些分片数据报并不总是有序的到达，而是处于一种无序状态，因此，接收主机便用此ID判断接收的这些分片数据报是否属于同一个数据流，然后再进行重组。

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？

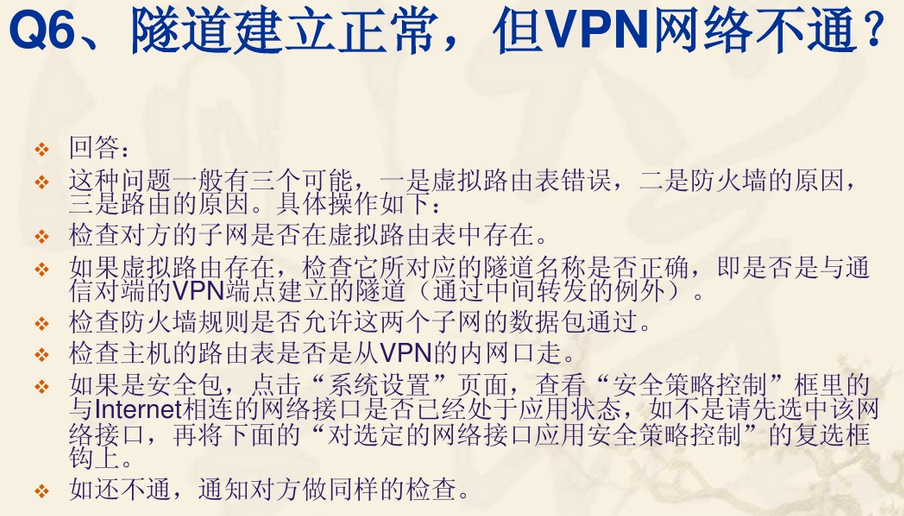
因VPN的实现需要对原有的IP或者是TCP/UDP数据进行封装，因此也就增加了数据包或数据帧的总长度，这样也就导致了VPN 承载的数据净荷值的减小，具体减少的数值与不同的VPN类型相关。

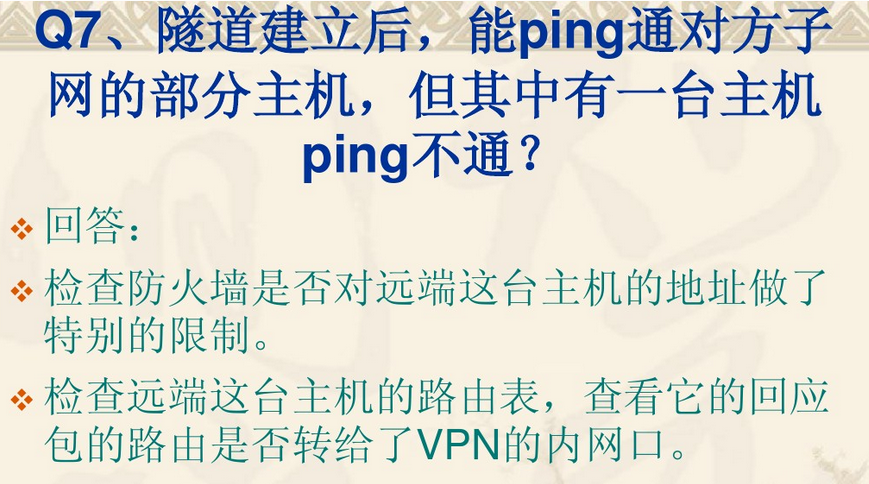
即：MTU=经VPN封装后数据包长度-VPN封装包头长度

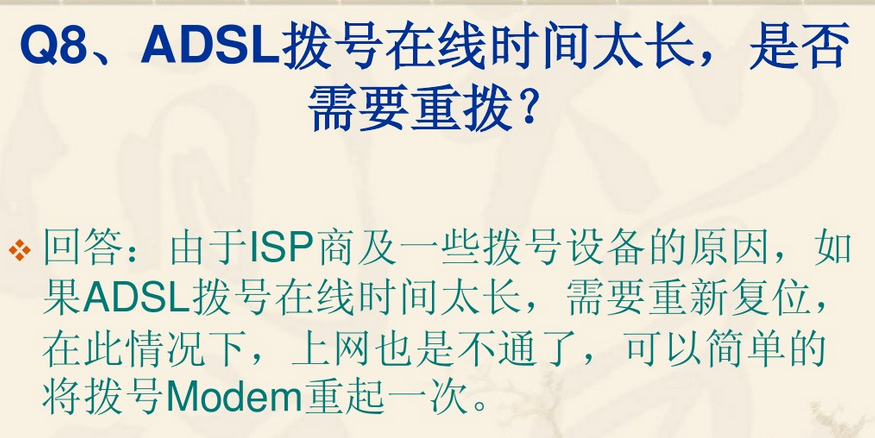
5、对于因MTU问题引起的应用系统故障现象有哪些？处理思路是什么？

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】：**

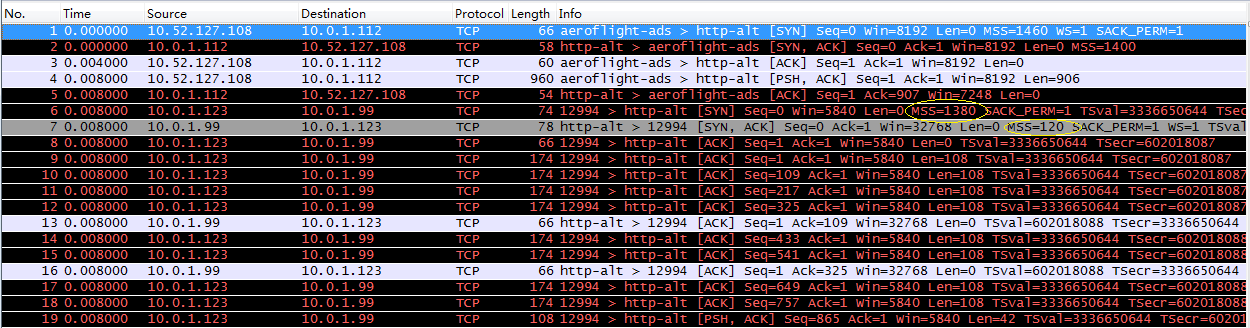
负载均衡设备位于客户端和真实服务器之间，一旦访问发生问题，在客户经过简单诊断后，负载均衡设备往往会成为首要被怀疑的对象。客户一般这样质疑：为什么我直接访问服务器没有问题，通过你的设备访问就不行了呢？ 质疑的确实有道理，但大多数事情往往不是非一即二这样简单，有很多东西都在互相影响，这就使得真相迷雾重重。

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：

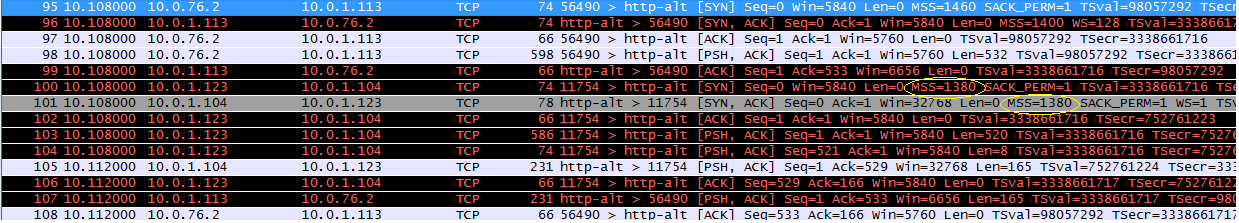
1) 10.52.127.108访问10.0.1.112

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)

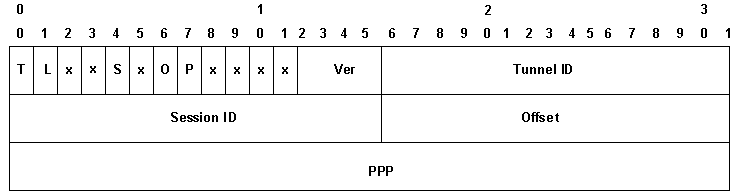
调整如何帮助减少一些相关问题。

## 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**

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。