解决 GRE 和 IPSEC 中的 IP 分段、MTU、MSS 和 PMTUD 问题

目录

简介

本文描述IP分段和路径最大传输单位发现(PMTUD)如何工作并且讨论介入PMTUD行为，当结合用IP隧道的不同的组合的一些方案。当前普遍使用IP隧道在互联网里带来介入IP分段和PMTUD对最前方的问题。

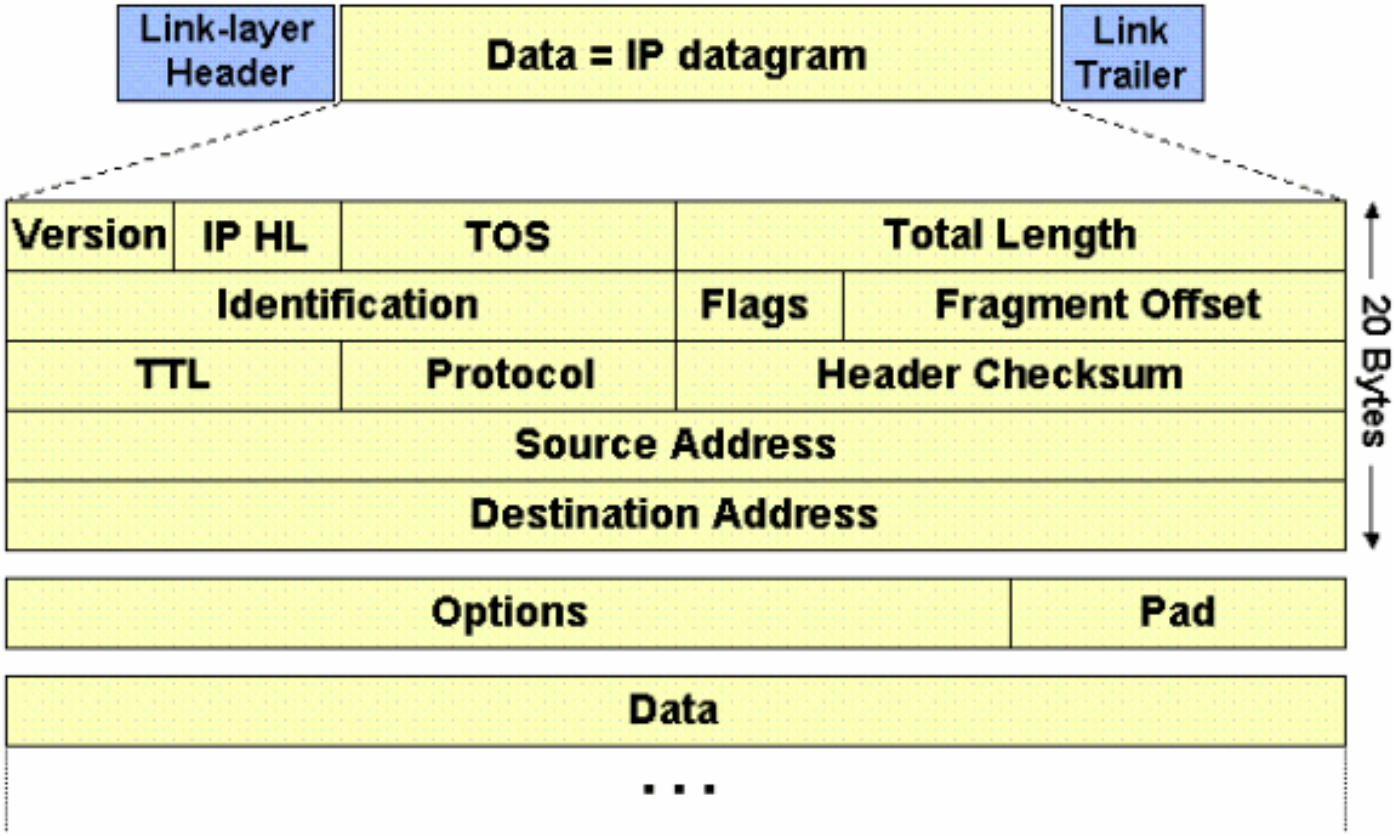
IP 分段与重组

IP 协议是专为用于各种传输链路而设计的。虽然IP数据包的最大长度是65535，多数传输链路强制执行一更加小的最大信息包长度限制，呼叫MTU。MTU 的值取决于传输链路的类型。因为允许路由器如所需要，分段IP数据包IP设计适应MTU差异。接收站对片段的重组负责回到原始大型的IP数据包。

IP 分段包括将数据报分为多个部分，可在稍后重组这些部分。IP 源、目标、标识、总长度和分段偏

移字段，以及 IP 报头中的“更多分段”标志和“不分段”标志均用于 IP 分段与重组。关于IP分段与重组的技工的更多信息，请参阅RFC 791 。

此镜像表示IP报头的布局。



识别是16个位并且是IP数据包的发送方分配的值帮助在数据包的片段的重组。

分段偏移为 13 位，指示分段在原始 IP 数据报中的所属位置。该值是 8 个字节的倍数。

在 IP 报头的标志字段中，有三位用于控制标志。请务必注意，“不分段”(DF) 位在 PMTUD 中发挥着中心作用，这是因为它确定是否允许对数据包进行分段。

位0保留和总是设置为0个。位1是DF位(0 = “可能分段”， 1 = “不片段”)。位 2 为 MF 位（0 =“最后一个分段”，1 =“更多分段”）。

值 位 0（保留）	位 1 (DF)	位 2 (MF)
0 0	5月	为时
1 0	不分段	更多

下图形显示分段示例。如果将所有 IP 分段的长度相加，所得的值将比原始 IP 数据报的长度大 60。总长度增加了 60 是因为另外创建了三个 IP 报头（第一个分段后的每个分段各对应一个 IP 报头）。

第一个分段的偏移为 0，此分段的长度为 1500；这包括已略作修改的原始 IP 报头所对应的 20 个字节。

第二个分段的偏移为 185 (185 x 8 = 1480)，这意味着此分段的数据部分从原始 IP 数据报的第 1480 个字节开始。此分段的长度为 1500；这包括另外为此分段创建的 IP 报头。

第三个分段的偏移为 370 (370 x 8 = 2960)，这意味着此分段的数据部分从原始 IP 数据报的第 2960

个字节开始。此分段的长度为 1500；这包括另外为此分段创建的 IP 报头。

第四个分段的偏移为 555 (555 x 8 = 4440)，这意味着此分段的数据部分从原始 IP 数据报的第 4440 个字节开始。此分段的长度为 700 个字节；这包括另外为此分段创建的 IP 报头。

只有在收到了最后一个分段时，才能确定原始 IP 数据报的大小。

通过最后一个分段的分段偏移 (555)，得出原始 IP 数据报中的数据偏移为 4440 个字节。如果再加上最后一个分段中的数据字节 (680 = 700 - 20)，这样您将获得 5120 个字节，这是原始 IP 数据报的数据部分。然后，加上 IP 报头的 20 个字节，即等于原始 IP 数据报的大小 (4440 + 680 + 20 = 5140)。

Original IP Datagram

Sequence	Identifier	Total Length	DF May / Don't	MF Last / More	Fragment Offset
0	345	5140	0	0	0

IP Fragments (Ethernet)

Sequence	Identifier	Total Length	DF May / Don't	MF Last / More	Fragment Offset
0-0	345	1500	0	1	0
0-1	345	1500	0	1	185
0-2	345	1500	0	1	370
0-3	345	700	0	0	555

IP 分段问题

存在下面几个致使 IP 分段不受欢迎的问题。分段 IP 数据报时，CPU 和内存开销将稍有增加。这对于发送器和接收器路径上的发送器和路由器是适用的。创建分段只涉及创建分段报头和将原始数据报复制到分段中。由于可以立即获取创建分段所需的所有信息，因此可以非常高效地完成该操作。

重组分段时，分段会导致接收者产生更大开销，原因是接收者必须为到达的分段分配内存，然后在收到所有分段后将它们重新组合为一个数据报。在主机上进行重组不会带来任何问题，因为主机拥有完成此任务所需的时间和内存资源。

但是，路由器的主要工作是尽快转发数据包，因此在路由器上进行重组的效率非常低。路由器不是为了将数据包保存任意时长而设计的。并且执行重组的路由器选择最大的缓冲区联机(18K)工作，因为没有方式认识原始IP数据包的大小，直到最后片段接收。

另一个分段问题介入已丢失片段如何被处理。如果 IP 数据报的某个分段被丢弃，则必须重新发送整个原始 IP 数据报，并且同样会对该数据报进行分段。以网络文件系统 (NFS) 为例进行说明。NFS，默认情况下，有读和写块大小8192，因此NFS IP/UDP数据包将是大约(包括NFS、UDP和IP头)的8500个字节。连接到以太网的发送站 (MTU 1500) 必须将此 8500 个字节的数据报分成 6 个部分；五个 1500 字节的分段和一个 1100 字节的分段。如果六个片段中的任一个丢弃由于阻塞链路，完整原始数据包将必须被重新传输，因此意味着六个片段将还必须创建。如果此链路丢弃六个数

据包中的其中一个数据包，那么可通过此链路传输任何 NFS 数据的可能性将非常低，这是因为每 NFS 8500 字节的原始 IP 数据报中将至少丢弃一个 IP 分段。

过滤或操作根据Layer4的数据包的防火墙(L4)通过在数据包的第七层(L7)信息也许有正确地处理IP段的麻烦。如果IP段有故障，防火墙也许阻拦??液初始分段，因为他们不传播将匹配数据包过滤器的信息。这意味着接收主机无法重组原始 IP 数据报。如果将防火墙配置为允许非初始分段包含不足以正确匹配过滤器的信息，则可能会发生通过防火墙进行非初始分段攻击的情况。并且，一些根据L4的网络设备(例如内容交换机引擎)直接数据包通过L7信息，和，如果数据包跨过多个片段，然后设备也许有强制执行其策略的麻烦。

避免IP分段：TCP MSS 用途及其工作原理

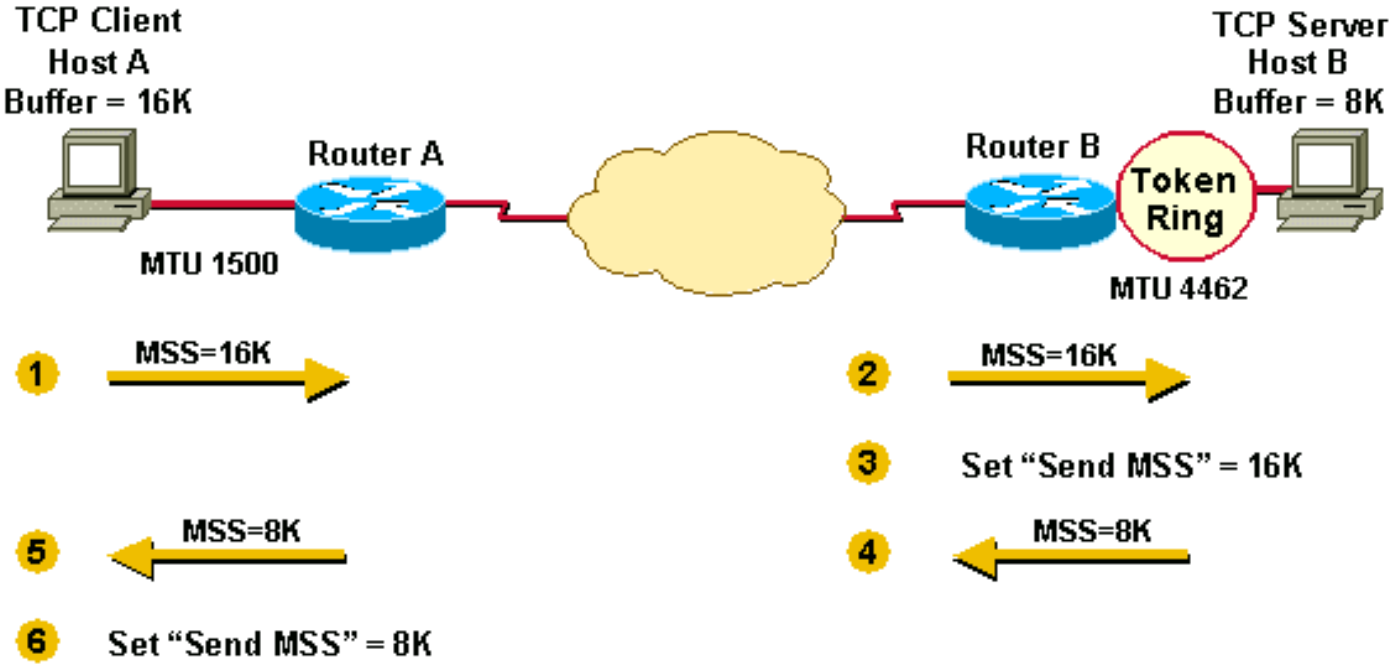
TCP 最大数据段大小 (MSS) 定义一台主机愿意接受的单一 TCP/IP 数据报中的最大数据量。此TCP/IP数据包也许被分段在IP层。MSS 值仅作为 TCP SYN 数据段中的一个 TCP 报头选项发送。TCP 连接的每一端都会向另一端报告其 MSS 值。与普遍看法相反的是，不会在主机之间协商 MSS 值。发送主机需要将单个 TCP 数据段中的数据大小限制为小于或等于接收主机报告的 MSS 的值。

最初，MSS 表示接收站上需要分配的缓冲区的大小（大于或等于 65496K），以便能够存储单个 IP 数据报内包含的 TCP 数据。MSS是TCP接收器愿意接收的最大数据分段(大块)。此 TCP 数据段最大可为 64K（即 IP 数据报最大大小），可以在 IP 层上进行分段，以通过网络传输到接收主机。接收主机将重组 IP 数据报，然后再将完整的 TCP 数据段传递给 TCP 层。

下面显示的两三个方案MSS值如何设置并且用于限制TCP数据段大小，并且， IP数据包大小。

方案 1 说明了 MSS 的最初实现方式。主机 A 的缓冲区为 16K，主机 B 的缓冲区为 8K。这些主机发送和接收其各自的 MSS 值，并调整其发送 MSS 以便彼此发送数据。注意主机A和主机B比发送MSS将必须分段大于接口MTU的IP数据包，但是仍然较少，因为TCP堆叠可能通过16K或8K字节的数据在堆叠下到IP。以主机 B 为例，可以对数据包进行两次分段，一次分段在令牌环 LAN 上执行，另一次在以太网 LAN 上执行。

场景 1



1. 主机 A 将其 MSS 值 16K 发送到主机 B。

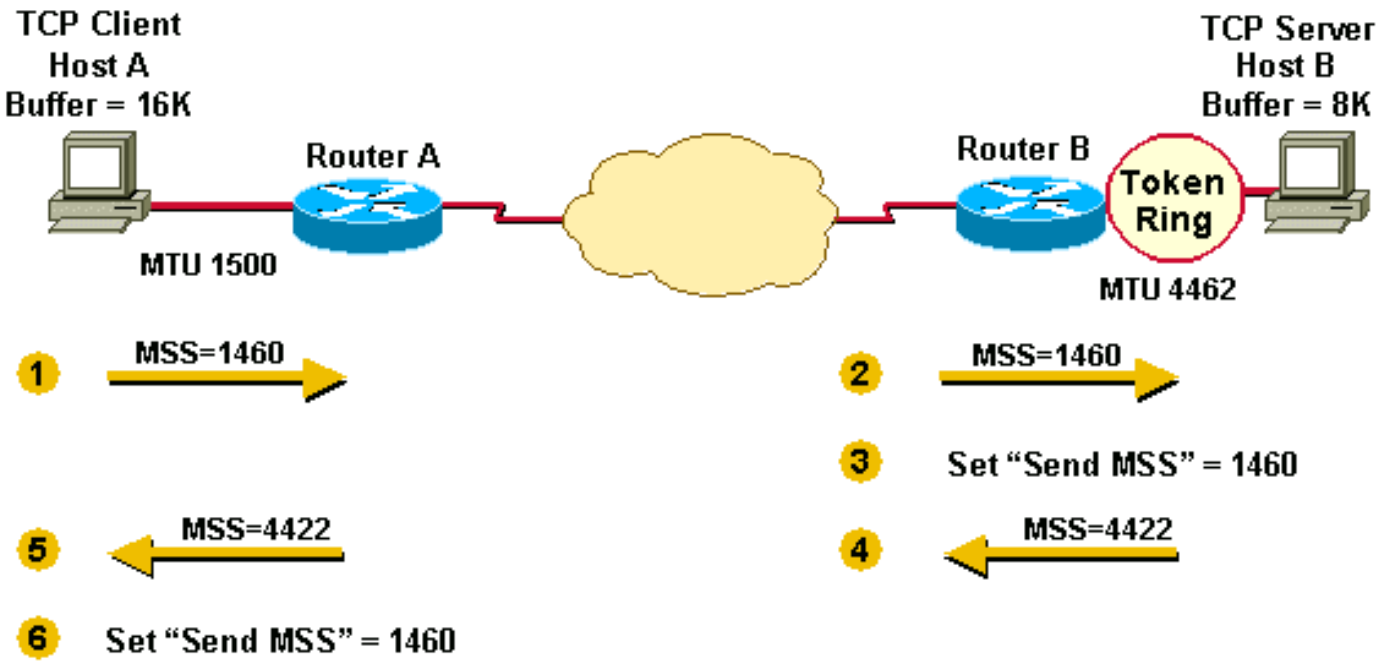
2. 主机 B 收到来自主机 A 的 MSS 值 16K。
3. 主机 B 将其发送 MSS 值设置为 16K。
4. 主机 B 将其 MSS 值 8K 发送到主机 A。
5. 主机 A 收到来自主机 B 的 MSS 值 8K。
6. 主机 A 将其发送 MSS 值设置为 8K。

为了帮助避免在 TCP 连接端点发生 IP 分段，MSS 值的选项已更改为最小缓冲区大小和传出接口的 MTU (-40)。MSS 数值比 MTU 数值小 40 字节，这是因为 MSS 仅为 TCP 数据大小，而不包括 20 字节的 IP 报头和 20 字节的 TCP 报头。MSS 基于默认报头大小；发送方堆叠必须减去IP报头的适当的值和TCP报头从属于什么TCP或Ip options使用。

现在，MSS 的工作方式是：各主机将首先比较其传出接口 MTU 和自己的缓冲区，并选择最小的值作为要发送的 MSS。然后，主机将比较收到的 MSS 大小和其自己的接口 MTU，并再次从两个值中选择较小的值。

方案2说明发送方采取的此额外步骤为了避免在本地和远程线路的分段。请注意每台主机如何将传出接口的 MTU 纳入考虑范畴（在主机互相发送其 MSS 值之前），以及这如何帮助避免分段。

场景 2



1. 主机 A 比较其 MSS 缓冲区 (16K) 和其 MTU (1500 - 40 = 1460)，并使用较小的值作为 MSS (1460)，以便发送到主机 B。
2. 主机 B 收到主机 A 的发送 MSS (1460)，并将其与出站接口 MTU - 40 的值 (4422) 相比较。
3. 主机 B 将较小的值 (1460) 设置为 MSS，以便向主机 A 发送 IP 数据报。
4. 主机 B 比较其 MSS 缓冲区 (8K) 和其 MTU (4462 - 40 = 4422)，并使用 4422 作为 MSS，以便发送到主机 A。
5. 主机 A 收到主机 B 的发送 MSS (4422)，并将其与出站接口 MTU - 40 的值 (1460) 相比较。
6. 主机 A 将较小的值 (1460) 设置为 MSS，以便向主机 B 发送 IP 数据报。

1460 便是两台主机为彼此选择的发送 MSS 的值。TCP 连接两端的发送 MSS 的值通常相同。

在方案 2 中，由于主机考虑了两个传出接口 MTU，因此不会在 TCP 连接端点进行分段。如果数据包遇到其 MTU 值小于主机出站接口的 MTU 值的链路，那么在路由器 A 和路由器 B 之间的网络中，仍可以对数据包进行分段。

什么是 PMTUD？

TCP MSS如及早描述照料分段在TCP连接的两个终端，但是不处理有一条更加小的MTU链路在这两个终端之间的中部的案件。PMTUD在终端之间的路径开发为了避免分段。它用于动态确定从数据包源到其目标之间的路径中的最小 MTU。

> **注意**：TCP和UDP只支持PMTUD。其他协议不支持它。如果PMTUD在主机启用，并且几乎总是，从主机的所有TCP/IP或UDP数据包将有设置的DF位。

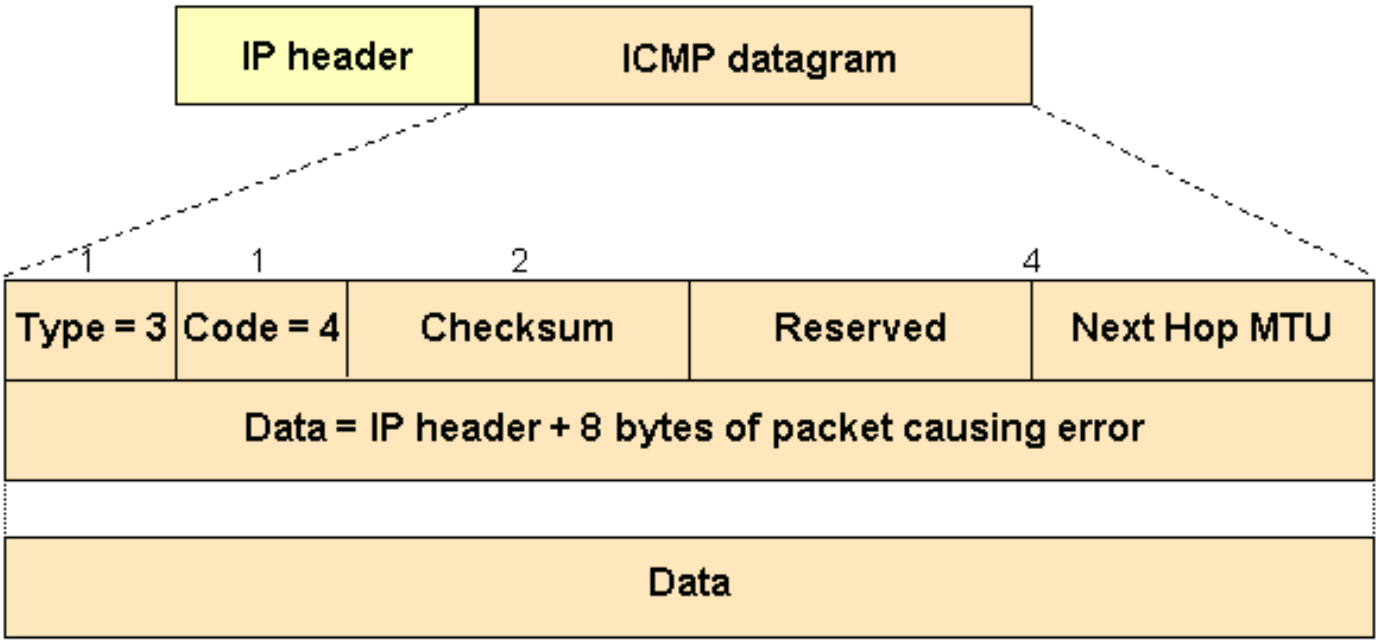
当主机发送有DF位集的时一个全双工MSS数据包， PMTUD降低连接的发送MSS值，如果获得信息数据包将要求分段。主机“通常记住”目的地的MTU值，因为在其路由表里创建一个“主机” (/32)条目与此MTU值。

如果路由器设法转发IP数据包，与DF位集，在比数据包的大小有一个更小的MTU的链路上，路由器将丢弃数据包并且返回互联网控制消息协议(ICMP) " Destination Unreachable "消息对此IP数据包来源，与指示“设置的分段需要的和DF”的代码(type3，代码4)。当源工作站收到 ICMP 消息时，将降低发送 MSS，并且当 TCP 重新传输该数据段时，它将使用更小的数据段大小。

这是您在路由器也许发现ICMP " fragmentation needed and DF set "消息的示例，在**debug ip icmp命令**打开后：

```
ICMP: dst (10.10.10.10) frag. needed and DF set
unreachable sent to 10.1.1.1
```

此图表显示“分段必要的和DF设置的” " Destination Unreachable "消息的ICMP报头格式。



每RFC 1191 ，返回ICMP信息指示的路由器“分段需要，并且设置的DF”应该在被标记“未使用”在ICMP规格RFC 792 ICMP另外的报头字段的低价位16个位包括该下一跳网络MTU 。

RFC1191 的早期实现未提供下一跳 MTU 信息。即使提供了此信息，某些主机也会忽略它。在本例中，RFC 1191 还包含一个表，该表中列出了应在 PMTUD 期间降低的 MTU 建议值。主机用于它为了快速到达在发送MSS的一个合理的值。

Plateau	MTU	Comments	Reference
	65535	Official maximum MTU	RFC 791
	65535	Hyperchannel	RFC 1044
65535			
32000		Just in case	
	17914	16Mb IBM Token Ring	ref. [6]
17914			
	8166	IEEE 802.4	RFC 1042
8166			
	4464	IEEE 802.5 (4Mb max)	RFC 1042
	4352	FDDI (Revised)	RFC 1188
4352 (1%)			
	2048	Wideband Network	RFC 907
	2002	IEEE 802.5 (4Mb recommended)	RFC 1042
2002 (2%)			
	1536	Exp. Ethernet Nets	RFC 895
	1500	Ethernet Networks	RFC 894
	1500	Point-to-Point (default)	RFC 1134
	1492	IEEE 802.3	RFC 1042
1492 (3%)			
	1006	SLIP	RFC 1055
	1006	ARPANET	BBN 1822
1006			
	576	X.25 Networks	RFC 877
	544	DEC IP Portal	ref. [10]
	512	NETBIOS	RFC 1088
	508	IEEE 802/Source-Rt Bridge	RFC 1042
	508	ARCNET	RFC 1051
508 (13%)			
	296	Point-to-Point (low delay)	RFC 1144
296			
68		Official minimum MTU	RFC 791

由于发送者和接收者之间的路径会动态发生变化，因此将在所有数据包上持续执行 PMTUD。每当发送者收到“无法分段”的 ICMP 消息时，它都会更新路由信息（PMTUD 存储在路由信息中）。

执行 PMTUD 时，可能会发生以下两种情况：

- 数据包可以一直发送到接收者，而不必分段。**注意**：为了使路由器能够保护 CPU 免受 DoS 攻击，它将发送的无法到达的 ICMP 消息数目限制为 2 条/秒。所以，在此上下文，如果有您期待的一网络环境路由器将需要回应超过两个ICMP消息(type=3，代码= 4)每秒(可以是不同的主机)，您会要禁用限制与**interface命令没有ip icmp速率限制不可得到的[df]的**ICMP消息。
- 发送器可以从接收器路径中的任何(或每一)跳沿路径获得ICMP "不能分段"信息。

PMTUD 在 TCP 流量的两个方向上独立执行。也许有在流一个方向的PMTUD触发其中一个终端站降低发送MSS的案件，并且另一端站点保持原始发送MSS，因为未曾发送足够大的IP数据包触发PMTUD。

下面方案 3 中描述的 HTTP 连接就是一个很好的示例。TCP客户端发送小数据包，并且服务器发送

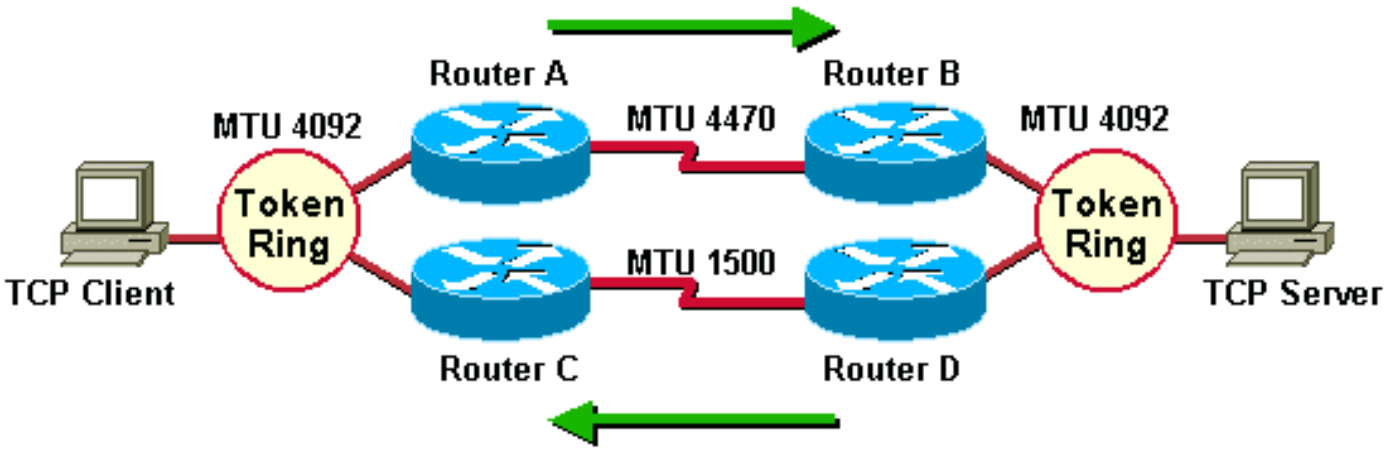
大数据包。在这种情况下，仅服务器的大数据包(非常地比576个字节)触发PMTUD。客户端的数据包相对较小（小于 576 字节），将不会触发 PMTUD，这是因为它们无需进行分段即可通过 576 MTU 链路。

场景 3

方案 4 显示的是不对称路由示例，在该示例中，其中一条路径的最小 MTU 小于另一路径的最小 MTU。当不同的路径采取发送和接收在两个终端之间时的数据不对称路由出现。在此方案中，PMTUD 仅在 TCP 流量的一个方向上触发对发送 MSS 的降低操作。从TCP客户端的流量服务器的流经路由器A和路由器B，而来自服务器到客户端的回程数据流流经路由器D和路由器C。当 TCP 服务器向客户端发送数据包时，PMTUD 将触发服务器以降低发送 MSS 的值，因为路由器 D 必须对 4092 字节的数据包进行分段，然后才能将其发送到路由器 C。

客户端，另一方面，不会接收与指示“设置的分段需要的和DF”的代码的一ICMP " Destination Unreachable "消息，因为路由器A不必须分段的信息包，当发送他们到服务器到路由器B。

场景 4



> **注意**：**ip tcp path-mtu-discovery** 命令用于对路由器启动的 TCP 连接（例如，BGP 和 Telnet）启用 TCP MTU 路径发现。

PMTUD 问题

有三种情况会中断 PMTUD，其中两种情况并不常见，而另一情况却经常发生。

- 路由器可能会丢弃数据包，并且不发送 ICMP 消息。(不常见)
- 路由器能生成和发送ICMP信息，但是ICMP信息由一路由器或防火墙阻拦在此路由器和发送方之间。(共同性)
- 路由器可以生成和发送 ICMP 消息，但发送者忽略了该消息。(不常见)

以上三项中的第一项和第三项并不常见，并且通常由错误所致，但中间一项描述的问题却较为常见

。实现 ICMP 数据包过滤器的人员往往会阻止所有 ICMP 消息类型，而不仅仅阻止特定的 ICMP 消息类型。除“无法到达”或“超时”消息以外，数据包过滤器可以阻止所有 ICMP 消息类型。PMTUD 成功与否取决于到达 TCP/IP 数据包的发送者的 ICMP 无法到达消息。ICMP 超时消息对其他 IP 问题至关重要。这样数据包过滤器示例，实现在路由器显示此处。

```
access-list 101 permit icmp any any unreachable
access-list 101 permit icmp any any time-exceeded
access-list 101 deny icmp any any
access-list 101 permit ip any any
```

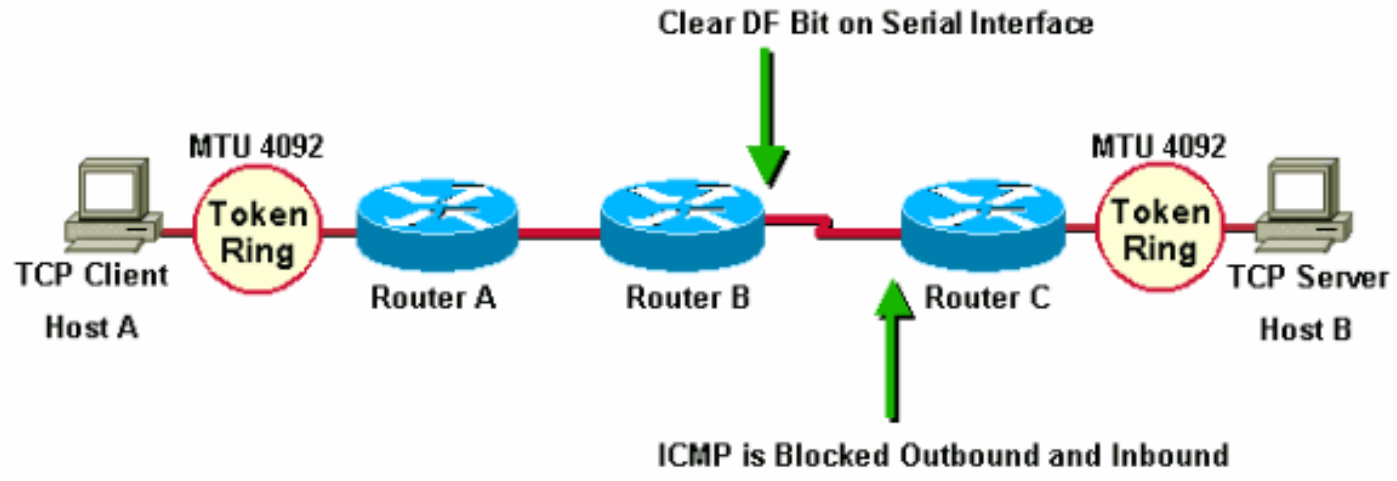
也可以使用其他技术来帮助排除完全阻止 ICMP 的问题。

- 清除路由器的DF位并且无论如何允许分段(这也许不是一个好想法，虽则。有关详细信息，请参阅 IP 分段问题）。
- 操作与interface命令**ip tcp adjust-mss <500-1460>**的TCP MSS选项值MSS。

在下个方案中，路由器A和路由器B是在同一管理域。路由器C是不可访问的并且阻塞ICMP，因此PMTUD是残破的。此情况的一应急方案是清楚两个方向的DF位在路由器B为了允许分段。这可以用策略路由执行。Cisco IOS® 软件版本 12.1(6) 和更高版本中提供了用于清除 DF 位的语法。

```
interface serial0
...
ip policy route-map clear-df-bit
route-map clear-df-bit permit 10
match ip address 111
set ip df 0

access-list 111 permit tcp any any
```



另一种选择是更改经过此路由器的 SYN 数据包上的 TCP MSS 选项值（Cisco IOS 12.2(4)T 和更高版本中可用）。这降低在TCP Syn信息包的MSS选项值，以便小于值(1460)在**ip tcp adjust-mss命令**。结果是TCP发送器将发送不大于该值的分段。IP信息包大小将是40个字节大(1500)比MSS值(1460个字节)为了占TCP报头(20个字节)和IP报头(20个字节)。

您可以使用 **ip tcp adjust-mss** 命令调整 TCP SYN 数据包的 MSS。此语法将降低在TCP分段的MSS值到1460。此命令将影响 serial0 接口上的入站和出站流量。

```
int s0
ip tcp adjust-mss 1460
```

由于 IP 隧道部署日益广泛，因此 IP 分段问题也越来越普遍。原因通道导致更多分段是，因为隧道封装添加“开销”到数据包的大小。例如，通用路由器封装(GRE)的新增内容添加24个字节到数据包，并且，在数据包也许需要被分段的此增加后，因为大于出站MTU。在后文中，您将了解隧道和 IP 分段可能导致的各种问题类型的示例。

需要 PMTUD 的常见网络拓扑

在网络环境中，如果中间链路的 MTU 小于终端链路的 MTU，则需要 PMTUD。存在这些较小的 MTU 链路的一些常见原因是：

- 与令牌环（或 FDDI）相连的终端主机之间存在以太网连接。令牌环(或FDDI)在末端的MTU比以太网MTU极大在中部。
- PPPoE（通常与 ADSL 配合使用）的报头需要 8 个字节。这使得以太网的有效 MTU 减小至 1492 (1500 - 8)。

隧道协议（如 GRE、IPsec 和 L2TP）还需要为它们各自的报头和报尾提供空间。这也会降低传出接口的有效 MTU。

在以下部分，隧道协议使用在两端主机之间PMTUD的影响被学习。三个上一个案件，此案件是多数复杂并且包括您也许在其他情况下发现的所有问题。

什么是隧道？

隧道是 Cisco 路由器上的一个逻辑接口，它提供了一种将乘客数据包封装在传输协议内的方法。设计此体系结构的目的是为了提供实现点对点封装方案的服务。隧道有这三个主要组件：

- 乘客协议（AppleTalk、Banyan VINES、CLNS、DECnet、IP 或 IPX）
- 载波协议-这些封装协议之一：GRE -思科的多协议承载协议。有关详细信息，请参阅 RFC 2784 和 RFC 1701。IP 中的 IP 隧道 - 有关详细信息，请参阅 RFC 2003。
- 传输协议 - 用于携带封装协议的协议

在此部分显示的数据包说明GRE是封装协议的IP隧道概念，并且IP是传输协议。此外，乘客协议也为 IP。在本案例中，IP 既是传输协议，也是乘客协议。

正常数据包

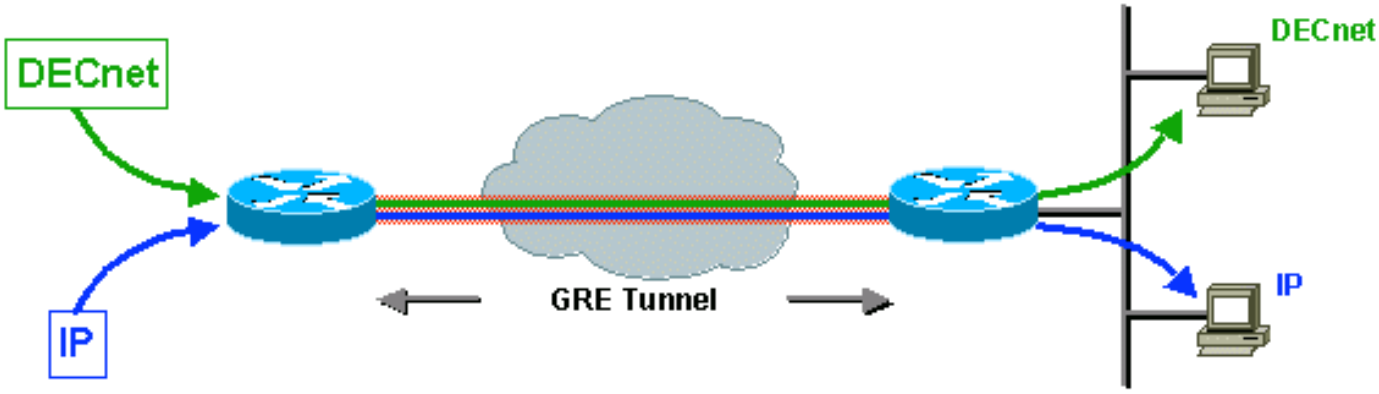
IP TCP Telnet

隧道数据包

IP GRE IP TCP Telnet

- IP 是传输协议。
- GRE 是封装协议。
- IP 是乘客协议。

下一示例显示了 IP 的封装方式，该示例采用 DECnet 作为乘客协议，并采用 GRE 作为载体。这说明了载体协议可以封装多个乘客协议的事实。



在存在两个被 IP 骨干网分离的不连续非 IP 网络的情况下，网络管理员也许会考虑建立隧道。如果不连续网络运行DECNet，管理员也许不要通过配置在骨干网的DECNet一起连接他们。因为这可能干涉IP网络的性能，管理员也许不要允许DECnet路由使用骨干网带宽。

一个可行的替代方法是在 IP 骨干网中对 DECnet 建立隧道。隧道封装DECnet信息包在IP里面，并且在骨干网间发送他们到封装删除的隧道终点，并且DECnet信息包可以路由到他们的目的地通过DECNet。

封装流量在另一份协议里面提供这些优点：

- 终端使用专用地址(RFC 1918)和骨干网不支持路由这些地址。
- 允许在 WAN 或 Internet 中建立虚拟专用网络 (VPN)。
- 通过一个单一协议的骨干网将不连续多协议网络连接在一起。
- 通过骨干网或 Internet 对流量进行加密。

对于本文的其余， IP使用作为乘客协议和IP作为传输协议。

有关隧道接口的注意事项

当建立隧道时，这些是考虑事项。

- GRE 隧道的快速交换功能是在 Cisco IOS 版本 11.1 中引入的，CEF 交换功能是在版本 12.0 中引入的。多点 GRE 隧道的 CEF 交换功能是在版本 12.2(8)T 中引入的。当支持，封装和解封装在隧道终点是缓慢的操作在Cisco IOS更早版本进程只交换。
- 对数据包建立隧道时，存在安全和拓扑问题。隧道可以绕过访问控制列表 (ACL) 和防火墙。如果通过防火墙建立隧道，您基本上就绕过了防火墙，无论您封装什么乘客协议。因此，建议在隧道端点中包含防火墙功能，以对乘客协议强制执行任何策略。
- 隧道也许制造与限制了计时器的传输协议的问题(例如， DECNet)由于加长的等待时间。
- 建立隧道在环境间用不同的速度链路，类似快速FDDI环和通过缓慢的9600 BPS电话线路，也许介绍重拨问题的数据包。一些乘客协议在混合媒体网络中的运行效率非常低下。
- 点对点隧道可能会用光物理链路的带宽。如果用在多点的路由协议指向通道，请记住每个隧道接口有一个带宽，并且通道运行的物理接口有一个带宽。例如，如果有 100 个隧道运行在一条 10 MB 的链路上，您可能希望将隧道带宽设置为 100 Kb。隧道的默认带宽是 9 Kb。
- 路由协议也许更喜欢在“实时”链路的一个通道，因为通道也许迷惑地看来是一跳链路用最便宜的路径，虽然实际上比另一个路径介入更多跳并且确实昂贵。通过正确配置路由协议可以消除此问题。您可能希望在隧道接口上运行不同的路由协议，而不是在物理接口运行的路由协议。
- 通过配置指向隧道目标的相应静态路由，可以避免递归路由问题。递归路由是指通往“隧道目标”的最佳路径是通过隧道本身。此情况促成隧道接口上下反弹。当有递归路由问题，您将看到此错误。`%TUN-RECURDOWN Interface Tunnel 0 temporarily disabled due to recursive routing`

路由器在隧道端点参与 PMTUD

当路由器作为隧道端点时，它将扮演两种不同的 PMTUD 角色。

- 路由器的第一个角色是作为主机数据包的转发者。对于 PMTUD 处理，路由器需要检查原始数据包的 DF 位和数据包大小，并在必要时采取相应操作。
- 在路由器将原始 IP 数据包封装在隧道数据包内之后，第二个角色开始发挥作用。在此阶段，路由器操作更多类似主机关于PMTUD和关于通道IP数据包。

通过查看什么让开始发生，当路由器在第一个角色操作，转发主机IP数据包的路由器，关于PMTUD。在路由器将主机 IP 数据包封装在隧道数据包内之前，此角色开始发挥作用。

如果路由器参与，因为主机数据包的转发器它将完成这些操作：

- 检查是否已设置 DF 位。
- 检查隧道可容纳的数据包大小。
- 分段（如果数据包太大，且未设置 DF 位），封装并发送分段；或
- 丢弃数据包（如果数据包太大，且设置了 DF 位），并向发送者发送 ICMP 消息。
- 封装（如果数据包并不太大）并发送。

一般地，有封装选择然后分段(发送两个封装片段)或分段然后封装(发送两个被封装的片段)。

描述IP数据包封装和分段的技工的一些示例和显示PMTUD交互作用和数据包横示例网络在此部分被选派的两个方案。

第一示例显示什么发生在数据包，当路由器(在隧道源)时在转发路由器角色操作。切记那处理PMTUD，路由器需要检查原始信息包的DF位和数据包大小和采取适当行为。本示例对隧道使用 GRE 封装。和能被看到， GRE在封装前执行分段。后面的示例显示在封装后进行分段的方案。

在示例 1 中，未设置 DF 位 (DF = 0)，并且 GRE 隧道 IP MTU 为 1476 (1500 - 24)。

示例 1

1. 转发路由器（位于隧道源中）从发送主机收到一个 1500 字节且清除了 DF 位 (DF = 0) 的数据报。此数据报由一个 20 字节的 IP 报头和一个 1480 字节的 TCP 负载组成。
2. 由于在增加 GRE 开销（24 字节）之后，对于 IP MTU 而言，数据包过大，因此，转发路由器会将数据报分为两个分别为 1476 字节（20 字节 IP 报头 + 1456 字节 IP 负载）和 44 字节（20 字节 IP 报头 + 24 字节 IP 负载）的分段，这样在添加 GRE 封装后，数据包不会大于传出物理接口 MTU。
3. 转发路由器向原始 IP 数据报的每个分段添加 GRE 封装，其中包括一个 4 字节的 GRE 报头和一个 20 字节的 IP 报头。这两个IP数据包当前有一个长度1500个和68个字节，并且这些数据包被看到作为个别的IP数据包，不作为片段。
4. 隧道目的地路由器从原始数据包的每个片段取消GRE封装，离开长度1476和24字节的两个IP段。该路由器将这些 IP 数据报分段单独转发到接收主机。
5. 接收主机将这两个分段重组为原始数据报。

方案 5 描述了转发路由器在网络拓扑环境中的角色。

在本例中路由器在转发路由器同一个角色操作，但是这次DF位设置(DF = 1)。

示例 2

1. 位于隧道源的转发路由器从发送主机收到一个 1500 字节且 DF = 1 的数据报。

2. 由于设置了 DF 位，并且数据报大小（1500 字节）大于 GRE 隧道 IP MTU (1476)，因此路由器将丢弃数据报，并向数据报源发送“需要分段但已设置 DF 位”的 ICMP 消息。ICMP 消息将警告发送者 MTU 为 1476。
3. 发送的主机接收ICMP信息，并且，当再发出原始数据时它将使用1476字节IP数据包。
4. 现在，此 IP 数据报的长度（1476 字节）等于 GRE 隧道 IP MTU 的值，因此路由器为此 IP 数据报添加 GRE 封装。
5. 接收路由器（位于隧道目标中）删除 IP 数据报的 GRE 封装，并将其发送到接收主机。

现在我们能查看什么发生，当路由器在第二个角色操作作为一台发送的主机关于PMTUD和关于通道IP数据包。回想一下，在路由器将原始 IP 数据包封装在隧道数据包内之后，此角色开始发挥作用。

> **注意**：默认情况下路由器不执行在生成的GRE隧道信息包的PMTUD。可以使用 **tunnel path-mtu-discovery** 命令对 GRE-IP 隧道数据包启用 PMTUD。

示例3显示发生了什么，当主机发送是足够小在GRE隧道接口的IP MTU内合适的IP数据包。在此情况下，可以设置或清除 DF 位（1 或 0）。GRE隧道接口没有**tunnel path-mtu-discovery命令**配置，因此路由器不执行在GRE-IP数据包的PMTUD。

示例 3

1. 位于隧道源的转发路由器从发送主机收到一个 1476 字节的数据报。
2. 此路由器将上述 1476 字节的 IP 数据报封装在 GRE 内，以得到一个 1500 字节的 GRE IP 数据报。将清除 GRE IP 报头中的 DF 位 (DF = 0)。然后，此路由器将此数据包转发到隧道目标。
3. 假定隧道源和目标之间存在一台路由器，并且链路 MTU 为 1400。由于已清除 DF 位 (DF = 0)，此路由器不会对隧道数据包进行分段。请记住，本示例只对最外层的 IP 进行分段，因而 GRE、内部 IP 和 TCP 报头将仅显示在第一个分段中。
4. 隧道目标路由器必须重组 GRE 隧道数据包。
5. 重组 GRE 隧道数据包之后，路由器将删除 GRE IP 报头，并在其路径中发送原始 IP 数据报。

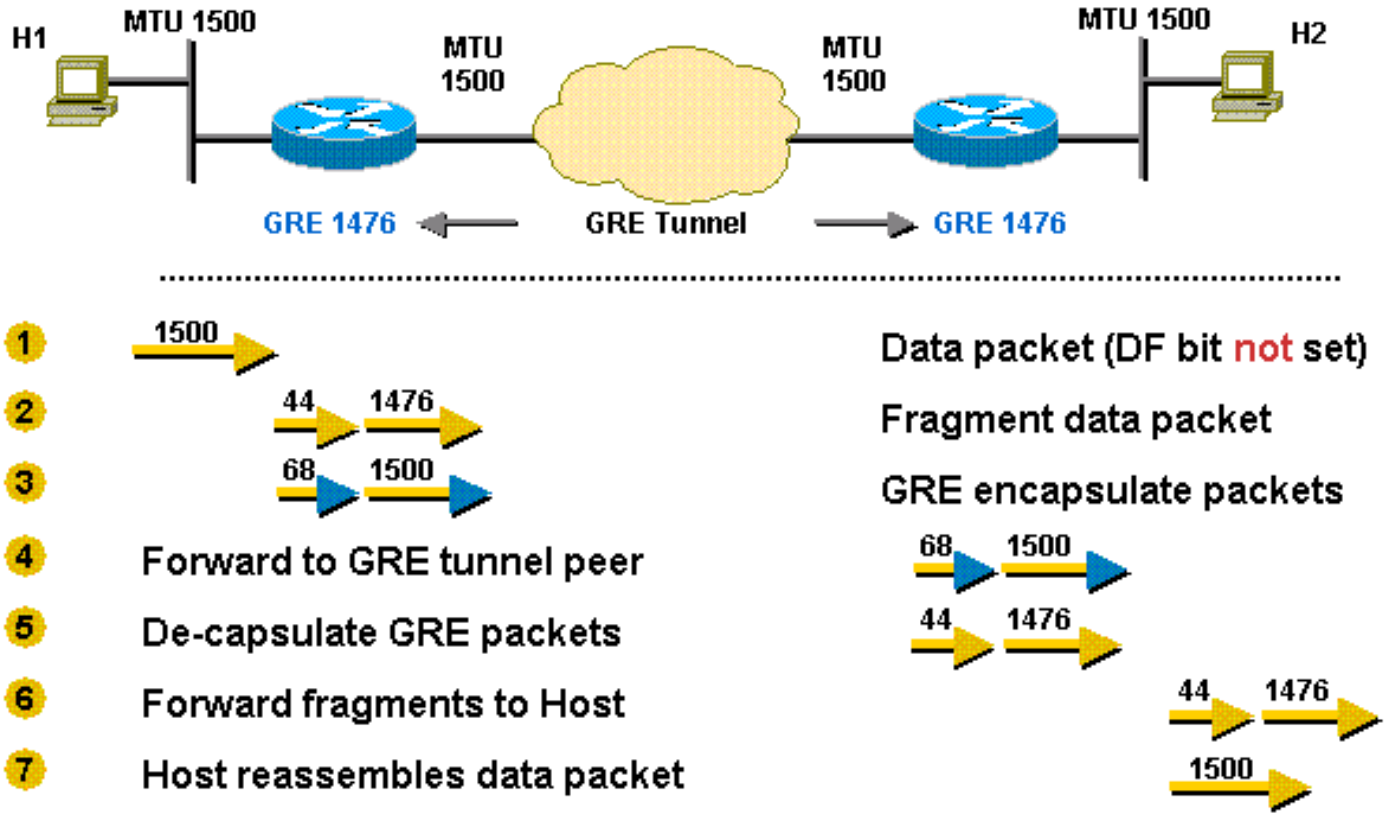
下一个示例显示发生了什么，当路由器在一台发送的主机的角色操作关于PMTUD和关于通道IP数据包。这次DF位设置(DF = 1)在原始IP报头和**tunnel path-mtu-discovery命令**配置，以便DF位从内在IP报头将复制到外面(GRE + IP)报头。

示例 4

1. 位于隧道源的转发路由器从发送主机收到一个 1476 字节且 DF = 1 的数据报。
2. 此路由器将上述 1476 字节的 IP 数据报封装在 GRE 内，以得到一个 1500 字节的 GRE IP 数据报。由于原始 IP 数据报设置了 DF 位，因此此 GRE IP 报头也将设置 DF 位 (DF = 1)。然后，此路由器将此数据包转发到隧道目标。
3. 同样，假定隧道源和目标之间存在一台路由器，并且链路 MTU 为 1400。由于设置了 DF 位 (DF = 1)，因此此路由器将不会对隧道数据包进行分段。此路由器必须丢弃数据包，并向隧道源路由器发送一条 ICMP 错误消息，这是因为隧道源路由器是数据包中的源 IP 地址。
4. 位于隧道源的转发路由器收到此 ICMP 错误消息，并将 GRE 隧道 IP MTU 降低为 1376 (1400 - 24)。当发送主机下次重新传输 1476 字节的 IP 数据包时，此数据包将过大，此路由器将向发送者发送一条 ICMP 错误消息，指明 MTU 值为 1376。当发送主机重新传输数据时，它将采用 1376 字节的 IP 数据包发送这些数据，因此该数据包将通过 GRE 隧道发送到接收主机。

方案 5

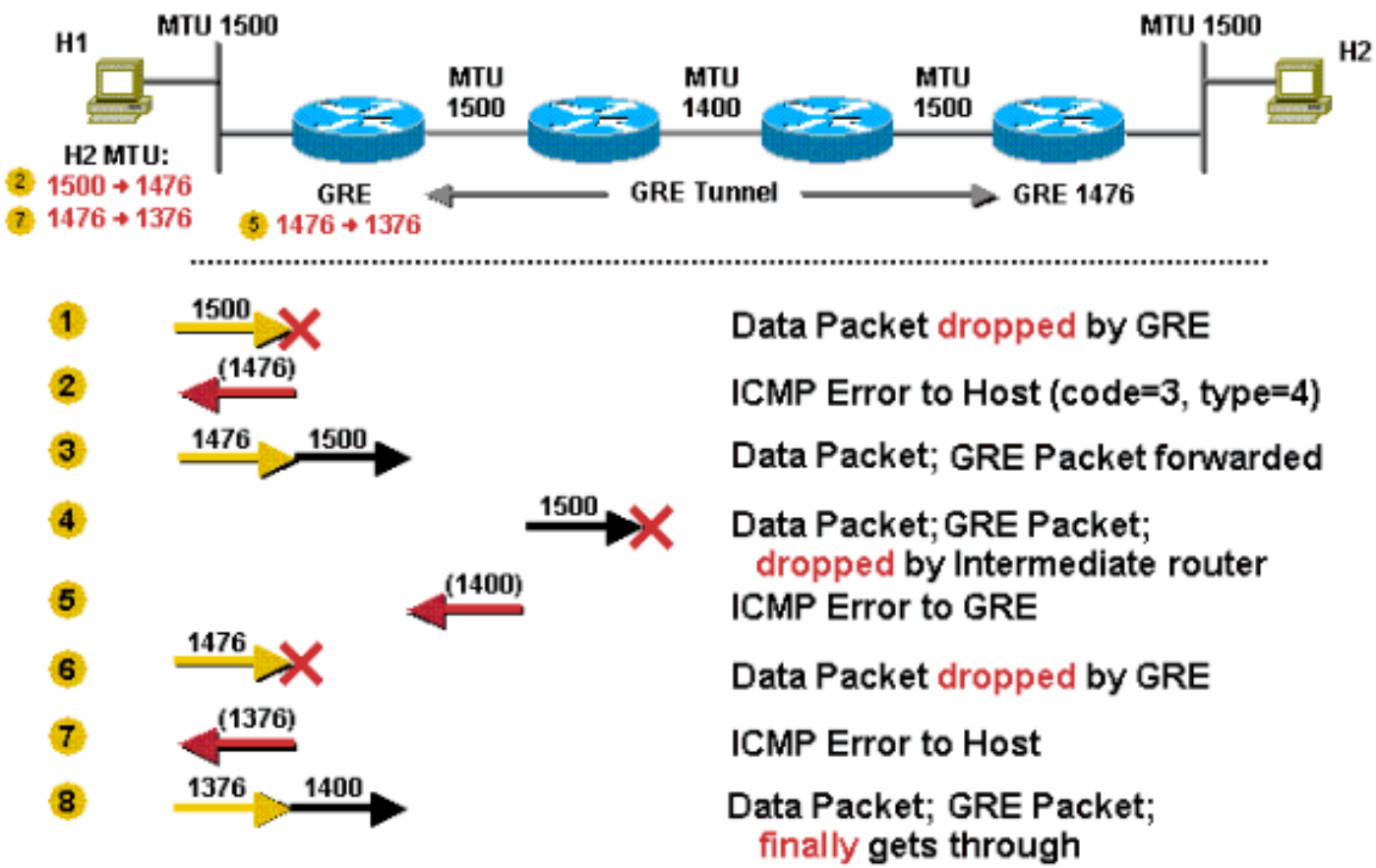
此方案说明 GRE 分段。请记住，应在封装 GRE 之前进行分段，然后对数据包执行 PMTUD，当使用 GRE 封装 IP 数据包时，不会复制 DF 位。在此方案中，未设置 DF 位。默认情况下，GRE 隧道接口 IP MTU 比物理接口 IP MUT 小 24 字节，因此 GRE 接口 IP MTU 为 1476。



1. 发送方发送1500字节数据包(20字节IP报头+ 1480字节的TCP有效载荷)。
2. 由于 GRE 隧道的 MTU 为 1476，因此，上述 1500 字节的数据包将被分为两个分别为 1476 字节和 44 字节的 IP 分段，每个分段将带额外的 24 字节的 GRE 报头。
3. 向每个 IP 分段添加到 24 字节的 GRE 报头。现在，两个分段分别为 1500 字节 (1476 + 24) 和 68 字节 (44 + 24)。
4. 包含两个IP段的GRE + IP信息包转发到GRE隧道对等`路由器。
5. GRE 隧道对等路由器将删除两个数据包中的 GRE 报头。
6. 此路由器将两个数据包转发到目标主机。
7. 目标主机将 IP 分段重新重组为原始 IP 数据报。

方案 6

此方案类似于方案5，但是这次DF位设置。在方案 6 中，将对路由器进行配置，以便使用 **tunnel path-mtu-discovery** 命令对 GRE + IP 隧道数据包执行 PMTUD，并将 DF 位从原始 IP 报头复制到 GRE IP 报头。如果路由器收到关于 GRE + IP 数据包的 ICMP 错误，则会减小 GRE 隧道接口上的 IP MTU。同样，请记住，默认情况下 GRE 隧道 IP MTU 设置为比物理接口 MTU 小 24 字节，因此，本示例中的 GRE IP MTU 为 1476。此外，请注意，GRE 隧道路径中存在一条 MTU 为 1400 的链路。



1. 路由器收到一个 1500 字节的数据包（20 字节 IP 报头 + 1480 字节 TCP 负载），然后丢弃此数据包。路由器丢弃数据包，因为大于IP MTU (1476)在GRE隧道接口。
2. 路由器向发送者发送一条 ICMP 错误，通知发送者下一跳 MTU 为 1476。主机将在其路由表中以目标主机路由的形式记录该信息。
3. 当重新发送数据时，发送主机采用 1476 字节作为数据包大小。GRE 路由器添加 24 字节的 GRE 封装，然后发送一个 1500 字节的数据包。
4. 该 1500 字节的数据包无法通过 1400 字节的链路，因此中间路由器将丢弃该数据包。
5. 中间路由器发送ICMP (type=3，代码= 4)对有下一跳的MTU GRE路由器1400。GRE 路由器将该数据包减小为 1376 (1400 - 24)，并在 GRE 接口上设置一个内部 IP MTU 值。仅当使用 **debug tunnel** 命令时，才会显示此更改；**show ip interface tunnel<#>** 命令的输出中不会显示此更改。
6. 下次主机再发出1476字节数据包， GRE路由器将丢弃数据包，因为大于当前IP MTU (1376)在GRE隧道接口。
7. GRE路由器将发送另一个ICMP (type=3、代码= 4)对有下一跳的MTU发送方1376和主机将更新其与新的值的当前信息。
8. 现在，主机将采用更小的 1376 字节的数据包重新发送该数据，GRE 将添加 24 字节的封装，并继续转发该数据包。这次数据包将使它到GRE隧道对等体，数据包将被解封装并且发送到目的地主机。**注意**：如果在本案例中，转发路由器上没有配置隧道 **path-mtu-discovery** 命令，并且DF位设置在GRF隧道转发的信息包中，那么主机1将向主机2继续发送TCP/IP信息包，但它们会在1400 MTUs链路的中途分段。此外，GRE 隧道对端必须重组这些数据包，然后才能解除封装和继续转发。

“纯”IPsec 隧道模式

IP安全协议是提供保密性、完整性和真实性给在间IP网络转接的信息的基于标准的方法。IPsec 提供 IP 网络层加密。IPsec 通过至少添加一个 IP 报头，从而增加了 IP 数据包的长度（隧道模式）。

已添加报头长度不同从属于IPSec配置模式，但是他们不超出~58个字节(封装安全有效载荷(ESP)和ESP验证(ESPauth))（封装安全负载 (ESP) 和 ESP 身份验证 (ESPauth)）。

IPsec 具有两种模式：隧道模式和传输模式。

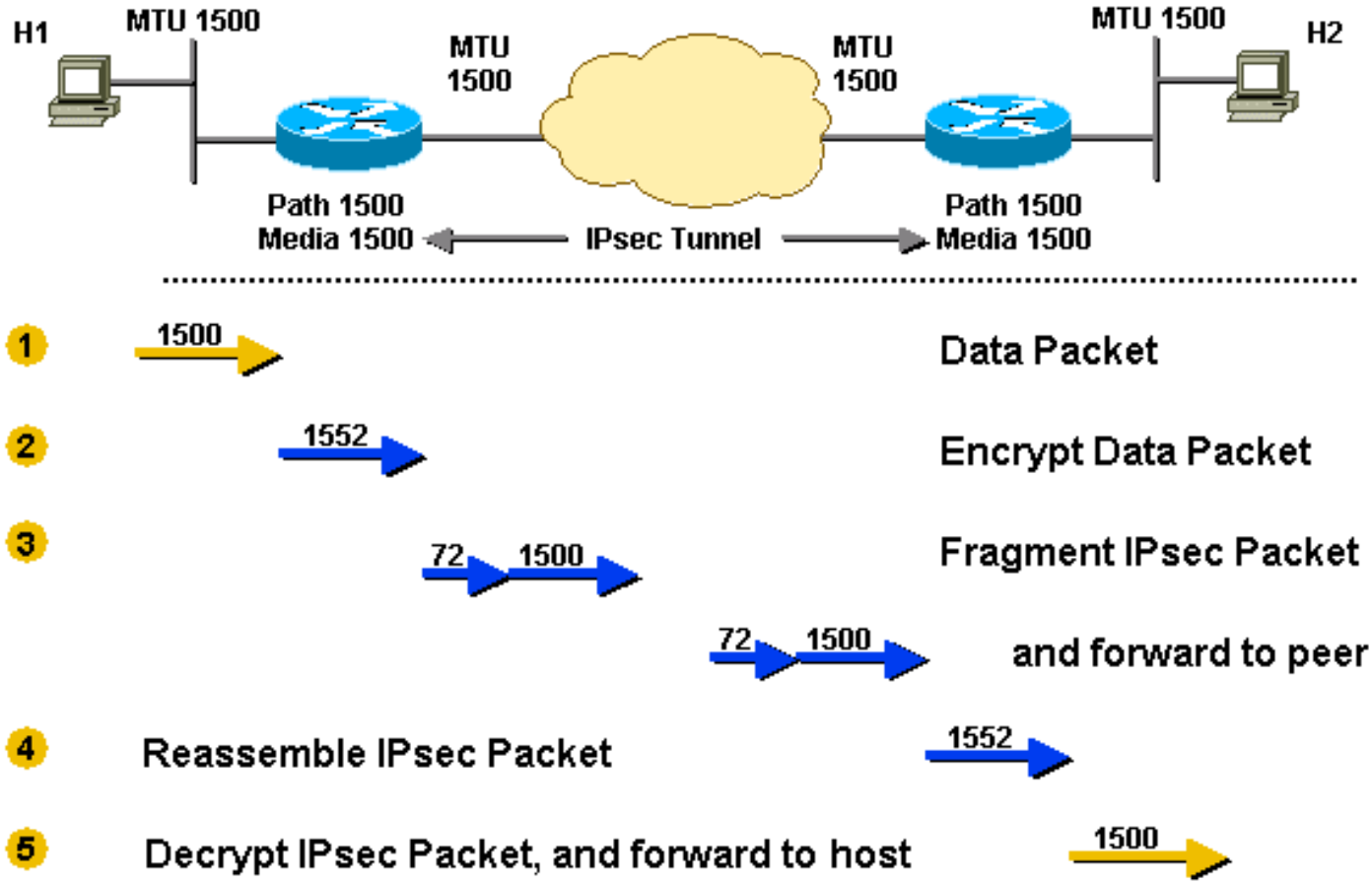
- 隧道模式是默认模式。使用隧道模式时，整个原始 IP 数据包处于受保护状态（已加密和/或已经过身份验证），并使用 IPSec 报头和报尾封装。然后一新的IP报头被加在前面到数据包，指定IPSec终点(对等体)作为源和目的。隧道模式可用于所有单播 IP 流量，如果 IPsec 为来自 IPSec 对端后面的主机的流量提供保护，则必须使用隧道模式。例如，在一个受保护网络中的主机通过一对 IPsec 对端向另一受保护网络中的主机发送数据包的虚拟专用网络 (VPN) 中，应使用隧道模式。在有VPN的情况下，IPsec "隧道"通过加密IPSec同位路由器之间的数据流来保护主机之间的IP数据流。
- 使用传输模式（在传输定义中使用 **mode transport** 子命令配置）时，只有原始 IP 数据包的负载才处于受保护状态（已加密和/或已经过身份验证）。该负载使用 IPSec 报头和报尾进行封装。除将 IP 协议字段更改为 ESP (50) 以外，原始 IP 报头将保持不变，原始协议值保存在 IPsec 报尾中，以便在数据包解密时进行恢复。仅当受保护的 IP 流量位于 IPSec 对端自身之间、数据包上的源和目标 IP 地址与 IPSec 对端地址相同时，才能使用传输模式。通常情况下，仅当首先使用另一隧道协议（如 GRE）封装 IP 数据包，然后使用 IPsec 来保护 GRE 隧道数据包时，才能使用 IPsec 传输模式。

IPsec 始终对数据包及其自己的数据包执行 PMTUD。存在可用于修改 IPsec IP 数据包的 PMTUD 处理的 IPSec 配置命令，IPsec 可在数据包 IP 报头中清除、设置 DF 位，或者将 DF 位从数据包 IP 报头复制到 IPsec IP 报头。该功能称为“DF 位覆盖功能”。

> **注意**：在使用 IPsec 执行硬件加密时，您一定希望在封装之后避免分段。根据所采用的硬件，硬件加密法可以为您提供大约 50 Mbs 的吞吐量，但如果对 IPsec 数据包进行分段，您将损失 50-90% 的吞吐量。导致该损失的原因在于，分段的 IPSec 数据包将执行进程交换以便重组，然后会传递到硬件加密引擎以进行解密。上述吞吐量损失会使硬件加密吞吐量降至软件加密的性能水平 (2-10 Mbs)。

方案 7

此方案描述了 IPsec 分段的作用方式。在此方案中，整个路径上的 MTU 为 1500。在此方案中，未设置 DF 位。

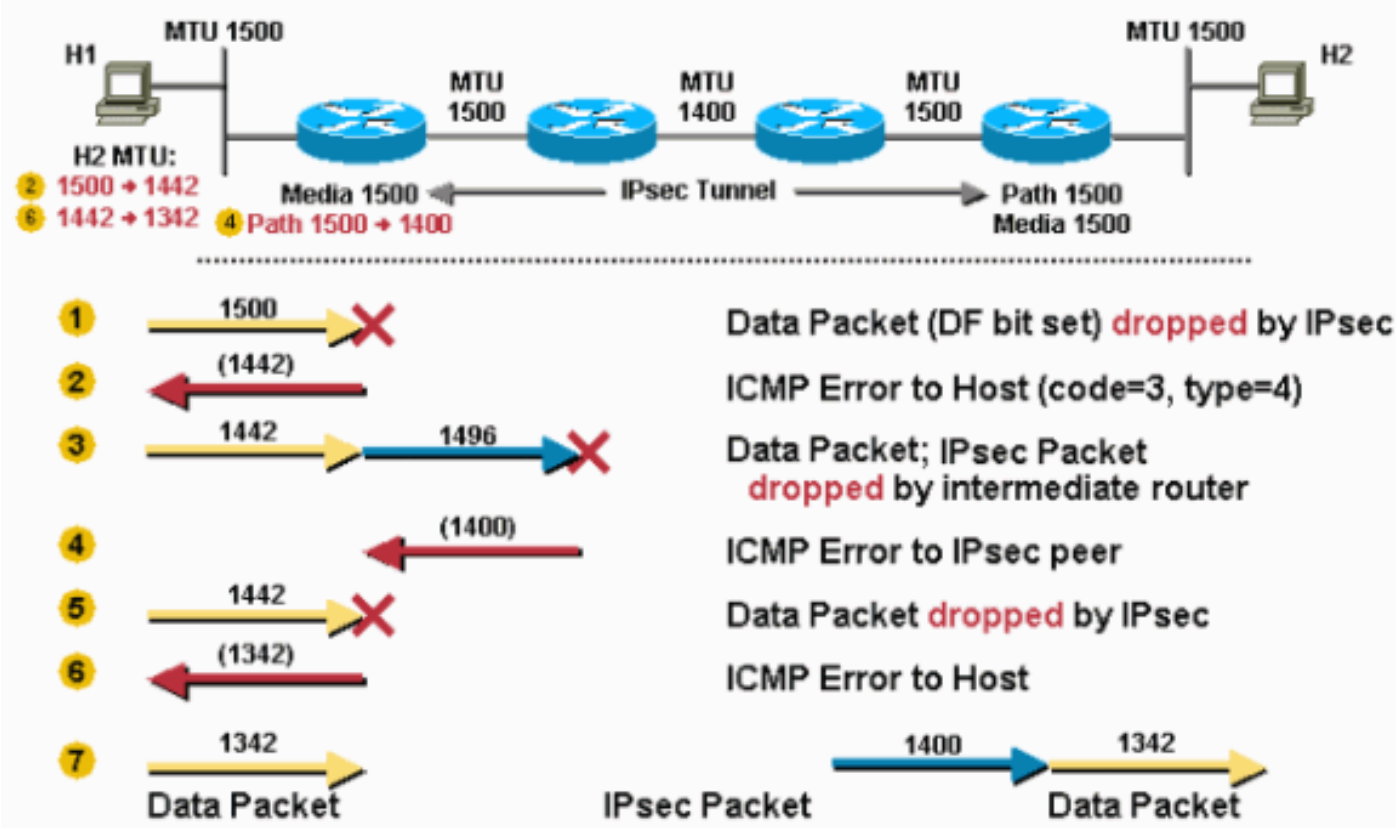


1. 路由器收到一个发往主机 2 且包含 1500 字节的数据包（20 字节 IP 报头 + 1480 字节 TCP 负载）。
2. 该 1500 字节的数据包使用 IPsec 进行加密，并增加了 52 字节的开销（IPSec 报头、报尾和额外的 IP 报头）。现在，IPsec 需要发送一个 1552 字节的数据包。由于出站 MTU 为 1500，因此必须对此数据包进行分段。
3. 将为此 IPsec 数据包创建两个分段。分段期间，将为第二个分段添加一个额外的 20 字节的 IP 报头，从而产生一个 1500 字节的分段和一个 72 字节的 IP 分段。
4. IPSec 隧道对等路由器收到分段，删除额外的 IP 报头，并将 IP 分段重新组合为原始的 IPsec 数据包。然后，IPsec 对此数据包进行解密。
5. 最后，路由器将 1500 字节的原始数据包转发到主机 2。

方案 8

此方案与方案 6 相似，不同之处在于此方案在原始数据包中设置了 DF 位，并且在 IPSec 隧道对端之间的路径中，有一条链路的 MTU 比其他链路的 MTU 小。此方案说明 IPSec 对等路由器如何执行两种 PMTUD 角色，如路由器在隧道端点参与 PMTUD 部分所述。

在此方案中，您将看到如何将 IPsec PMTU 更改为较小值（出于分段需要）。请记住，当 IPsec 对数据包进行加密时，将从内部 IP 报头将 DF 位复制到外部 IP 报头。媒体 MTU 和 PMTU 值存储在 IPSec 安全关联 (SA) 中。媒体 MTU 基于出站路由器接口的 MTU，PMTU 基于 IPSec 对端之间的路径上的最小 MTU。请记住，IPsec 会先封装/加密数据包，然后再尝试对数据包进行分段。



1. 路由器收到一个 1500 字节的数据包并丢弃该数据包，这是因为在增加 Ipsec 开销之后，数据包将大于 PMTU (1500)。
2. 路由器向主机 1 发送一条 ICMP 消息，并通知该主机下一跳 MTU 为 1442 (1500 - 58 = 1442)。当使用 IPsec ESP 和 ESPauth 时，此 58 字节是最大 Ipsec 开销。实时IPsec开销比此值可能是多达7个字节较少。主机 1 通常在其路由表中以目标（主机 2）主机路由的形式记录该信息。
3. 主机 1 将针对主机 2 将其 PMTU 减小至 1442，因此主机 1 在向主机 2 重新传输数据时，将发送更小（1442 字节）的数据包。路由器收到 1442 字节的数据包，IPsec 添加 52 字节的加密开销，这样，产生的 IPSec 数据包便为 1496 字节。由于此数据包的报头中设置了 DF 位，因此，采用 1400 字节 MTU 链路的中间路由器将丢弃此数据包。
4. 丢弃数据包的中间路由器向 IPSec 数据包的发送者（第一个路由器）发送一条 ICMP 消息，通知它下一跳 MTU 为 1400 字节。此值记录在 SA IPsec PMTU 中。
5. 主机 1 下次重新传输上述 1442 字节的数据包时（未收到有关此数据包的确认），IPsec 将丢弃此数据包。再次路由器将丢弃数据包，因为在头顶上IPsec，当添加到数据包，比PMTU (1400)将使变大。
6. 路由器向主机 1 发送一条 ICMP 消息，通知它下一跳 MTU 现在为 1342。(1400 - 58 = 1342)。主机 1 将再次记录此信息。
7. 主机 1 再次重新传输数据时，它将采用更小的数据包 (1342)。此数据包无需分段即可通过 IPSec 隧道发送到主机 2。

同时使用 GRE 与 IPsec

当IPsec用于为了加密GRE隧道时，分段的更加复杂的交互作用和PMTUD发生。IPsec和GRE如此被结合，因为IPsec不支持IP组播数据包，因此意味着您不能运行在IPSec VPN网络的一个动态路由协议。GRE 隧道支持多播，因此可以首先使用 GRE 隧道封装 GRE IP 单播数据包中的动态路由协议多播数据包，然后再使用 IPsec 进行加密。执行此操作时，通常在传输模式下在 GRE 之上部署 Ipsec，这是因为 IPSec 对等体和 GRE 隧道端点（路由器）相同，传输模式将减少 20 字节的

IPsec 开销。

一个有趣的情形是：当一个 IP 数据包被拆分为两个分段，并使用 GRE 封装之后。在此种情况下，IPsec 会将其视为两个独立 GRE + IP 数据包。通常，在默认配置中，其中一个数据包将足够大，因此在加密之后需要对该数据包进行分段。IPSec 对等体必须在解密之前重组此数据包。发送路由器上的“两次分段”（一次在 GRE 之前，一次在 IPsec 之后）会延长等待时间，并降低吞吐量。此外，重组是进程交换过程，因此重组时接收路由器上将大量占用 CPU 资源。

通过在 GRE 隧道接口上将“ip mtu”设置为足够低，以便将来自 GRE 和 IPsec 的开销计算在内（默认情况下，GRE 隧道接口“ip mtu”设置为实际传出接口 MTU - GRE 开销字节），可以避免此类情况的发生。

此表列出假设每个的通道/的模式组合的建议的MTU值流出的物理接口有MTU 1500。

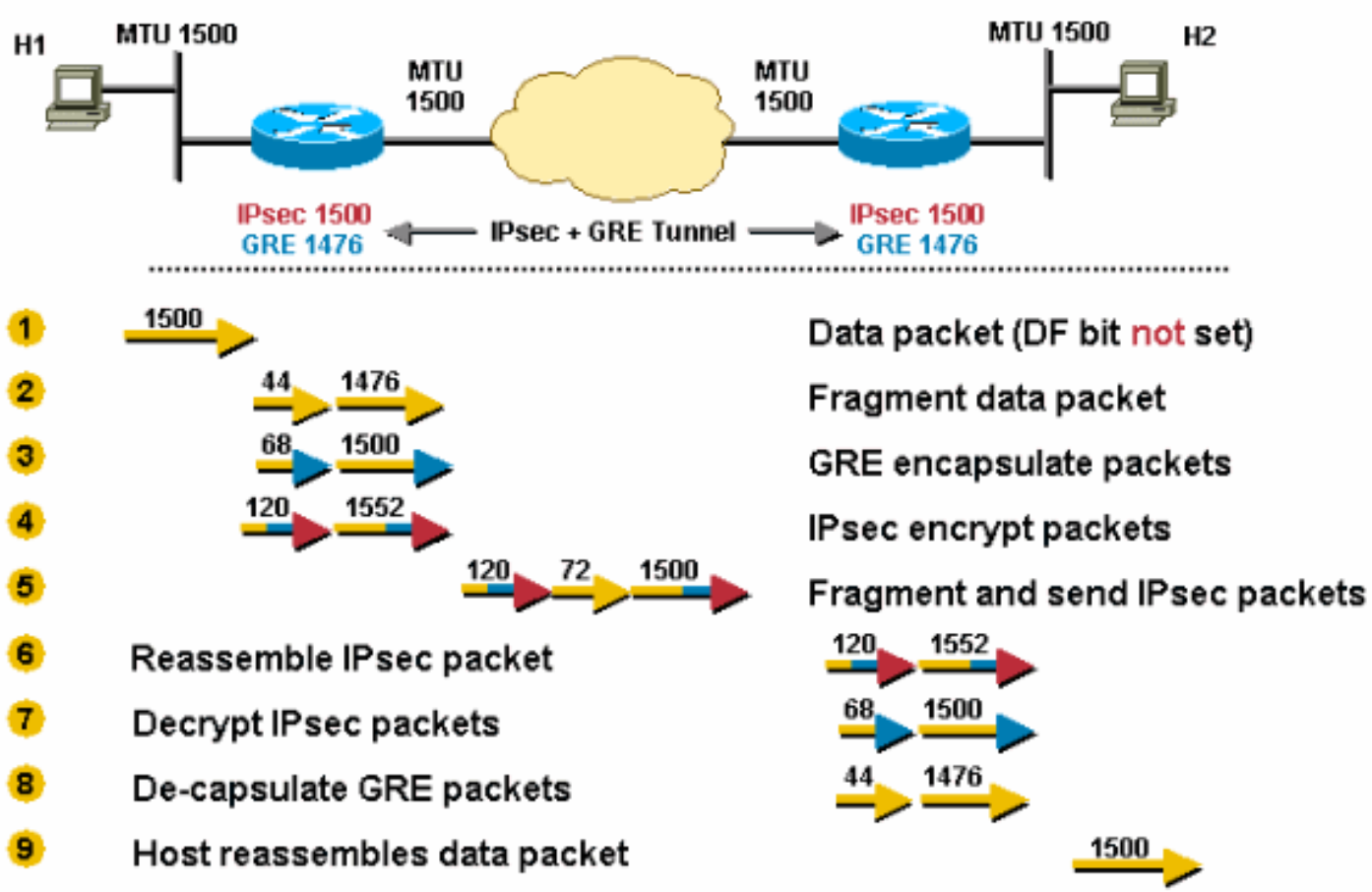
隧道组合	**需要的特定 MTU**	**建议的 MTU**
GRE + IPsec（传输模式）	1440 字节	1400 字节
GRE + IPsec（隧道模式）	1420 字节	1400 字节

> **注意**：建议的 MTU 值为 1400，它涵盖最常见的 GRE + IPsec 模式组合。并且，额外允许 20 或 40 字节的开销不会产生显著的负面影响。记住并设置一个值相对较容易，并且该值几乎涵盖了所有情况。

方案 9

在 GRE 之上部署 IPsec。传出物理 MTU 为 1500，IPsec PMTU 为 1500，GRE IP MTU 为 1476 (1500 - 24 = 1476)。因此，将对 TCP/IP 数据包进行两次分段，一次在 GRE 之前，一次在 Ipsec 之后。该数据包将在 GRE 封装之前进行分段，其中一个 GRE 数据包将在 IPSec 加密之后再次进行分段。

在 GRE 隧道上配置“ip mtu 1440”（IPsec 传输模式）或“ip mtu 1420”（IPSec 隧道模式），将消除该方案中可能要进行两次分段这一情况。

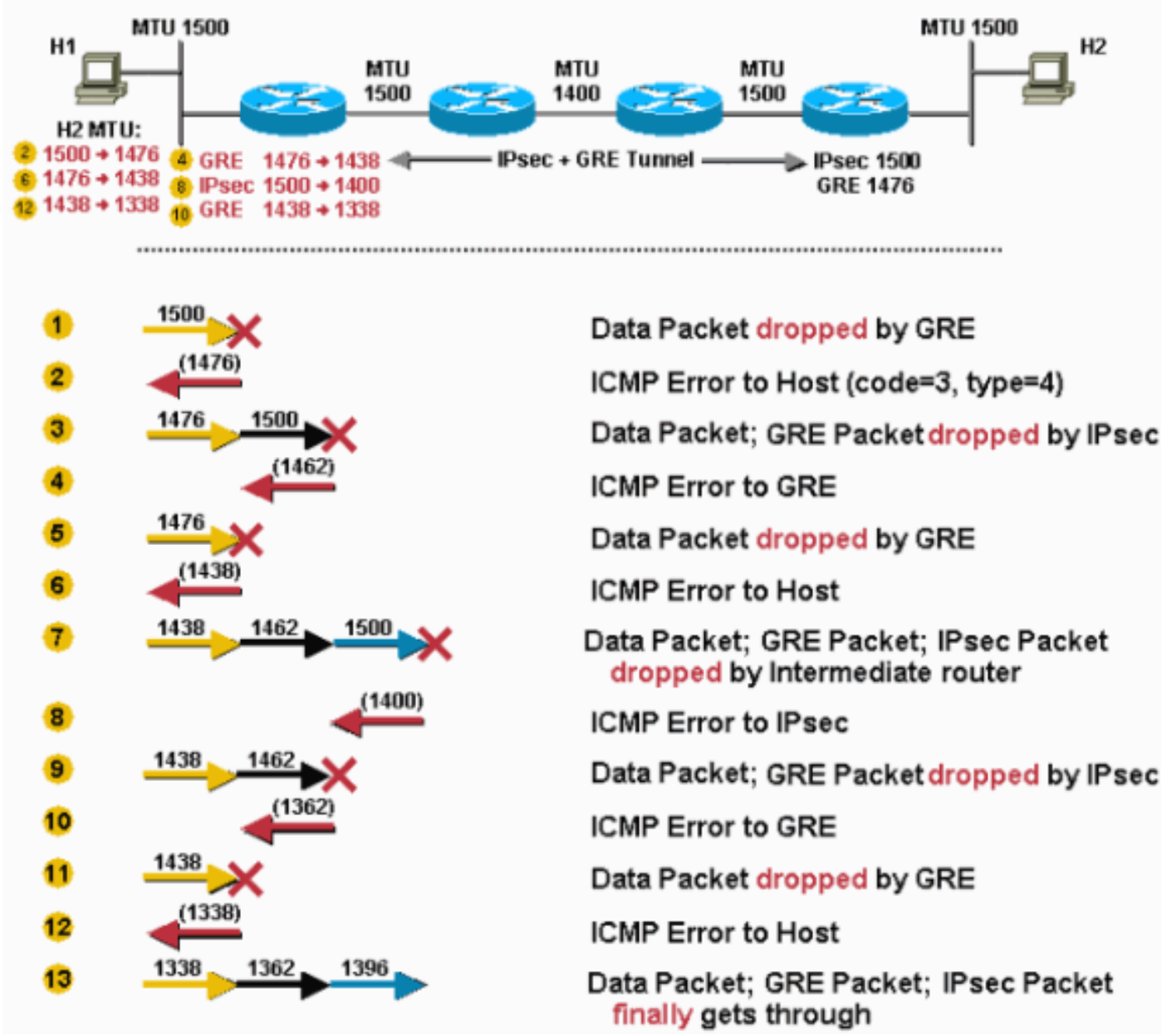


1. 路由器收到一个 1500 字节的数据报。
2. 封装之前，GRE 将 1500 字节的数据包分为两个部分，分别为 1476 字节 (1500 - 24 = 1476) 和 44 字节（24 字节数据 + 20 字节 IP 报头）。
3. GRE 封装 IP 分段，并向每个数据包添加 24 个字节。这将产生两个分别为 1500 字节 (1476 + 24 = 1500) 和 68 字节 (44 + 24) 的 GRE + IPSec 数据包。
4. IPsec加密两数据包，添加52 bye (IPsec隧道模式)封装开销到其中每一，为了给1552字节和120-byte数据包。
5. 1552字节IPSec信息包由路由器断片化，因为大于出站MTU (1500)。1552 字节的数据包分为两个部分，即一个 1500 字节的数据包和一个 72 字节的数据包（52 字节的“负载”加上第二个分段中额外的 20 字节的 IP 报头）。这三个分别为 1500 字节、72 字节和 120 字节的数据包将转发到 IPsec + GRE 对等体。
6. 接收路由器重新组装两个IPsec片段(1500个字节和72个字节)为了得到原始1552字节IPsec+GRE数据包。不必对 120 字节的 IPsec + GRE 数据包执行任何操作。
7. IPsec解码1552字节和120-byte IPsec+GRE数据包为了得到1500字节和68字节GRE数据包。
8. GRE解封装1500字节和68字节GRE数据包为了获得1476字节和44字节IP数据包片段。这些 IP 数据包分段将转发到目标主机。
9. Host2重新组装这些IP段为了获得原始1500字节IP数据包。

方案 10 与方案 8 相似，不同之处在于方案 10 的隧道路径中存在更小 MTU 的链路。这是从主机 1 发送到主机 2 的第一个信息包的“最坏情况”。在完成此方案的最后一步之后，主机 1 将针对主机 2 设置正确的 PMTU，对于主机 1 和主机 2 之间的 TCP 连接，所有设置都可以正常运行。主机 1 和其他主机（可通过 IPsec + GRE 隧道访问）之间的 TCP 流量只需完成方案 10 中的最后三个步骤。

在此方案中， **tunnel path-mtu-discovery命令**在GRE隧道配置，并且DF位在于Host1产生的TCP/IP数据包设置。

方案 10



1. 路由器收到一个 1500 字节的数据包。由于设置了 DF 位，并且在增加 GRE 开销（24 字节）之后数据包大小超过出站接口“ip mtu”，因此 GRE 无法对该数据包进行分段或转发，并丢弃此数据包。
2. 路由器发送ICMP信息对Host1为了告诉它下一跳MTU是1476 (1500 - 24 = 1476)。
3. 主机 1 针对主机 2 将其 PMTU 更改为 1476，并在重新传输数据包时发送更小大小。GRE 封装该数据包，并将 1500 字节的数据包传递给 IPsec。由于 GRE 从内部 IP 报头复制了其中设置的 DF 位，并且带有 IPsec 开销（最大数目为 38 字节），此时数据包过大，不能通过物理接口进行转发，因此 IPsec 将丢弃此数据包。
4. IPsec发送ICMP信息对表明的GRE下一跳MTU是1462个字节(因为最大数量38字节为加密和IP将被添加在头顶上)。GRE在隧道接口上将值1438 (1462 - 24)记录为"ip mtu"。**注意**：将在内部存储对值所做的上述更改，而不会在 **show ip interface tunnel<#>** 命令的输出中显示。仅当改用 **debug tunnel** 命令时，才会显示此更改。
5. 主机 1 下次重新传输 1476 字节的数据包时，GRE 将丢弃此数据包。
6. 表明的路由器发送ICMP信息对Host1 1438是下一跳MTU。
7. 主机 1 针对主机 2 减小其 PMTU，并重新传输 1438 字节的数据包。此时，GRE 将接受并封装此数据包，并将其传递给 Ipsec 进行加密。IPSec 数据包转发到中间路由器并被丢弃，这是因为中间路由器的出站接口 MTU 为 1400。
8. 告诉它的中间路由器发送ICMP信息对IPsec下一跳MTU是1400。此值由 IPsec 记录在相关

IPsec SA 的 PMTU 值中。
9. 当主机 1 重新传输 1438 字节的数据包时，GRE 会对其进行封装，并将其传递给 IPsec。由于已将自己的 PMTU 更改为 1400，因此 IPsec 将丢弃此数据包。
10. IPsec发送表明的ICMP错误对GRE下一跳MTU是1362，并且GRE记录值1338内部地。
11. 当主机1重新传输原始信息包时(因为没有收到确认)，GRE将丢弃它。
12. 指示的路由器发送ICMP信息对Host1下一跳MTU是1338 (1362个- 24个字节)。主机 1 针对主机 2 将其 PMTU 减小至 1338。
13. 主机1转发1338字节信息包，同时它可以最终到达主机2。

其他建议

如果在同一台路由器上配置了 GRE 和 IPsec，则在隧道接口上配置 **tunnel path-mtu-discovery** 命令可以帮助 GRE 和 Ipsec 交互。请记住，如果未配置 **tunnel path-mtu-discovery** 命令，GRE IP 报头中将始终会清除 DF 位。这允许将被分段的，即使GRE IP数据包封装的数据IP报头有DF位设置，通常不会允许数据包将被分段。

如果**tunnel path-mtu-discovery命令**在GRE隧道接口配置，这将发生。

1. GRE 从数据 IP 报头将 DF 位复制到 GRE IP 报头。
2. 如果在 GRE IP 报头中设置了 DF 位，并且在采用 IPSec 加密之后，数据包对物理传出接口上的 IP MTU 显得“过大”，IPsec 将丢弃此数据包，并通知 GRE 隧道减小其 IP MTU 大小。
3. IPsec执行其自己的数据包和，如果IPsec PMTU更改(如果减少)，然后IPsec的PMTUD不立即通知GRE，但是，当另一“太大”数据包来周到时，然后进程在步骤2发生。
4. GRE的IP MTU现在更小，因此它将丢弃带有DF位设置的IP信息包（现在过大），并向发送主机发送ICMP信息。

tunnel path-mtu-discovery 命令有助于 GRE 接口动态设置其 IP MTU，而不是使用 ip mtu 命令静态设置。实际上，建议同时使用这两个命令。**ip mtu** 命令用于为 GRE 和 Ipsec 开销提供空间（相对于本地物理传出接口 IP MTU）。如果 IPSec 对等体之间的路径中存在一条具有更小 IP MTU 的链路，使用 **tunnel path-mtu-discovery** 命令可以进一步减小 GRE 隧道 IP MTU。

如果配置了 GRE + IPSec 隧道的网络中出现 PMTUD 问题，可以采取以下所列措施。

此列表开始与最理想的解决方案。

- 解决问题由于不工作的PMTUD，通常是由路由器或防火墙引起的阻塞ICMP。
- 在隧道接口上使用 **ip tcp adjust-mss** 命令，以使路由器减小 TCP Syn 数据包中的 TCP MSS 值。这有助于两台终端主机（TCP 发送者和接收者）使用足够小的数据包，以便无需执行 PMTUD。
- 在路由器的入口接口上使用策略路由，配置路由映射以在到达 GRE 隧道接口之前清除数据 IP 报头中的 DF 位。这将允许在 GRE 封装之前对数据 IP 数据包进行分段。
- 在 GRE 隧道接口上增大“ip mtu”，使其等于出站接口 MTU。这将允许使用 GRE 对数据 IP 数据包进行封装，而不必首先对其进行分段。然后，将使用 IPsec 对 GRE 数据包加密，并对其进行分段以通过物理出站接口。在此种情况下，您不必在 GRE 隧道接口上配置 **tunnel path-mtu-discovery** 命令。由于会以进程交换模式在 IPSec 对等体上对 IP 数据包进行重组，因此这会大幅降低吞吐量。

相关信息

- IP 路由支持页
- IPSec（IP 安全协议）支持页
- IPSec顶上的计算器(请计算与IPSec封装协议的数据包大小)
- RFC 1191 路径 MTU 发现
- RFC 1063 IP MTU 发现选项
- RFC 791 Internet 协议
- RFC 793 传输控制协议
- RFC 879 TCP 最大数据段大小和相关主题
- RFC 1701 通用路由封装 (GRE)
- RFC 1241 Internet 封装协议方案
- RFC 2003 IP 中的 IP 封装
- 技术支持 - Cisco Systems