解决 GRE 和 IPsec 中的 IPv4 分段、MTU、MSS 和 PMTUD 问题

目录

简介

背景信息

IPv4 分段和重组

- IPv4 分段相关问题
- 避免 IPv4 分段：TCP MSS工作原理
  - 示例 1
  - 示例 2

PMTUD 是什么？

- 示例 3
- 示例 4
- PMTUD 相关问题
- 需要 PMTUD 的常见网络拓扑

隧道

- 有关隧道接口的注意事项
- 在隧道终端作为 PMTUD 参与者的路由器
  - 示例 5
  - 示例 6

纯 IPsec 隧道模式

- 示例 7
- 示例 8

GRE 与 IPv4sec 协同工作

- 示例 9
- 示例 10

更多建议

相关信息

简介

本文档介绍 IPv4 分段和路径最大传输单元发现 (PMTUD) 的工作方式。

背景信息

还讨论了涉及 PMTUD 在与 IPv4 隧道的不同组合结合时的行为的场景。

IPv4 分段和重组

尽管 IPv4 数据报的最大长度为 65535 字节，但大多数传输链路强制执行更小的最大数据包长度限

制（即 MTU）。MTU 值取决于传输链路。

设计 IPv4 时已考虑 MTU 的差异，因为它允许路由器根据需要对 IPv4 数据报进行分段。

接收站（在此情景中）负责将分段重组为原始的全尺寸IPv4数据报。

IPv4 分段功能将数据报拆分为之后可以重组起来的多个片段。

IPv4 报头中的 IPv4 源地址、目的地址、标识符、总长度和分段偏移量字段以及“更多分段”(MF) 和“不分段”(DF) 标志都被用于 IPv4 分段和重组。

有关 IPv4 分段和重组机制的更多信息，请参阅 RFC 791。

下图显示了 IPv4 报头的布局。

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

标识符是 IPv4 数据报发送方分配的 16 位值，它有助于重组数据报的分段。

分段偏移量字段长 13 位，表示分段在原始 IPv4 数据报中所属的位置。此值为 8 字节的倍数。

IPv4 报头的标志字段中有 3 个控制标志位。“不分段”(DF) 位确定是否允许对数据包进行分段。

位 0 为预留位，始终设置为 0。

位 1 为 DF 位（0 =“可以分段”，1 =“不分段”）。

位 2 为 MF 位（0 =“最后一个分段”，1 =“更多分段”）。

价值	位 0（保留）	位 1 (DF)	位 2 (MF)
0	0	5 月	最后一页
1	0	不分段	更多

如果将 IPv4 分段的长度相加，则总长度将比原始 IPv4 数据报长 60 字节。

总长度之所以增加 60 字节是因为为第一个分段后的另外三个分段多创建了三个 IPv4 报头，每个分段一个。

第一个分段的偏移量为 0，分段长度为 1500 字节；其中有 20 字节用于略有修改的原始 IPv4 报头。

第二个片段的偏移量为185(185 x 8 = 1480);此分段的数据部分从原始IPv4数据报的1480字节开始。

此分段的长度为 1500；其中包括为此分段额外创建的 IPv4 报头。

第三个片段的偏移量为370(370 x 8 = 2960);此分段的数据部分从原始IPv4数据报的2960字节开始。

此分段的长度为 1500；其中包括为此分段额外创建的 IPv4 报头。

第四个分段的偏移量为 555 (555 x 8 = 4440)，这意味着此分段的数据部分从原始 IPv4 数据报的第 4440 字节后开始。

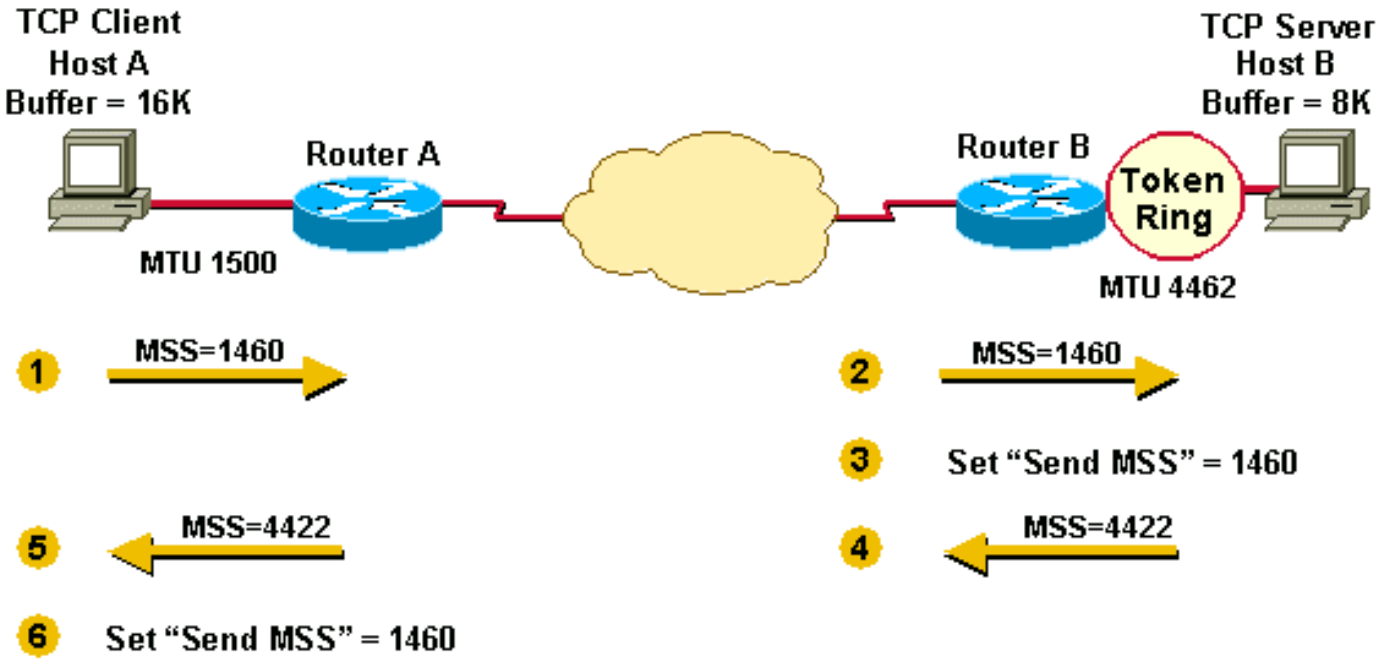
此分段的长度为 700 个字节；其中包括为此分段额外创建的 IPv4 报头。

只有当收到最后一个分段时，才可以确定原始 IPv4 数据报的大小。

最后一个分段中的分段偏移量 (555) 表示数据偏移量为原始 IPv4 数据报的 4440 字节。

最后一个分段的数据字节数之和 (680 = 700-20) 为 5120 字节，这就是原始 IPv4 数据报的数据部分。

再加上 IPv4 报头的 20 字节，等于原始 IPv4 数据报的大小 (4440 + 680 + 20 = 5140)，如下图所示。



IPv4 分段相关问题

IPv4 分段会导致对 IPv4 数据报进行分段时出现 CPU 和内存开销小幅增加。不管是对于发送端还是对于发送端与接收端之间的路由器来说，都会出现这种情况。

分段的创建涉及创建分段报头以及将原始数据报复制到分段中。

由于创建分段所需的所有信息均即时可用，因此分段创建可以高效完成。

由于在重组分段时，接收端必须为到达的分段分配内存，并在接收所有分段后，将其重新合并为一个数据报，因此分段会给接收端带来更多的开销。

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。

然而，对于主要任务是尽快转发数据包的路由器而言，分段重组却是一个非常低效的工作。

路由器不是为了将数据包保存任意时长而设计的。

执行重组的路由器会选择使用最大的可用缓冲区 (18K)，因为在收到最后一个分段之前，它无法确定原始 IPv4 数据包的大小。

另一个分段问题涉及到已丢弃分段的处理方式。

如果 IPv4 数据报的某个分段被丢弃，则整个原始 IPv4 数据报必须存在并且它也被分段。

网络文件系统 (NFS) 中会出现这种情况。NFS 的读写块大小为 8192 字节。

因此，NFS IPv4/UDP 数据报大约为 8500 字节（其中包括 NFS、UDP 和 IPv4 报头）。

连接到以太网(MTU 1500)的发送站必须将8500字节的数据报分段为六(6)段；五(5)个1500字节片段和一(1)个1100字节片段。

如果由于链路堵塞而丢弃了六个分段中的任何一个，则必须重新传输完整的原始数据报。这会导致创建另外六个分段。

如果此链路每传输六个数据包就丢弃其中一个数据包，则说明可通过此链路传输任何 NFS 数据的可能性很低，因为每个 NFS 8500 字节的原始 IPv4 数据报中，至少有一个 IPv4 分段会被丢弃。

根据第 4 层 (L4) 到第 7 层 (L7) 的信息对数据包进行过滤或操作的防火墙无法正确处理 IPv4 片段。

如果 IPv4 分段杂乱无序，防火墙会阻止非初始分段，因为它们未携带与数据包过滤器匹配的信息。

这意味着接收主机无法重组原始 IPv4 数据报。

如果将防火墙配置为允许非初始分段（这些分段未充分包含用于正确匹配过滤器的信息）通过，则可能通过防火墙发生非初始分段攻击。

内容交换引擎等网络设备根据 L4 至 L7 的信息来定向数据包，如果数据包跨越多个分段，则设备将难以执行其策略。

避免 IPv4 分段：TCP MSS工作原理

传输控制协议 (TCP) 最大分段大小 (MSS) 定义主机在单个 TCP/IPv4 数据报中接受的最大数据量。

此 TCP/IPv4 数据报可能会在 IPv4 层进行分段。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。MSS 值不在主机之间协商。

发送主机需要将单个 TCP 分段中的数据量限制为小于等于接收主机报告的 MSS 值。

最初，MSS 决定着接收站上分配用于存储单个 IPv4 数据报所含 TCP 数据的缓冲区的大小（大于等于 65496 字节）。

MSS 是指 TCP 接收端愿意接受的最大数据分段。此 TCP 分段最大可以为 64K，并可以在 IPv4 层进行分段，以便传输到接收主机。

在将完整的 TCP 分段传送至 TCP 层之前，接收主机会重组 IPv4 数据报。

如何设置 MSS 值以及如何使用它来限制 TCP 分段和 IPv4 数据报的大小。

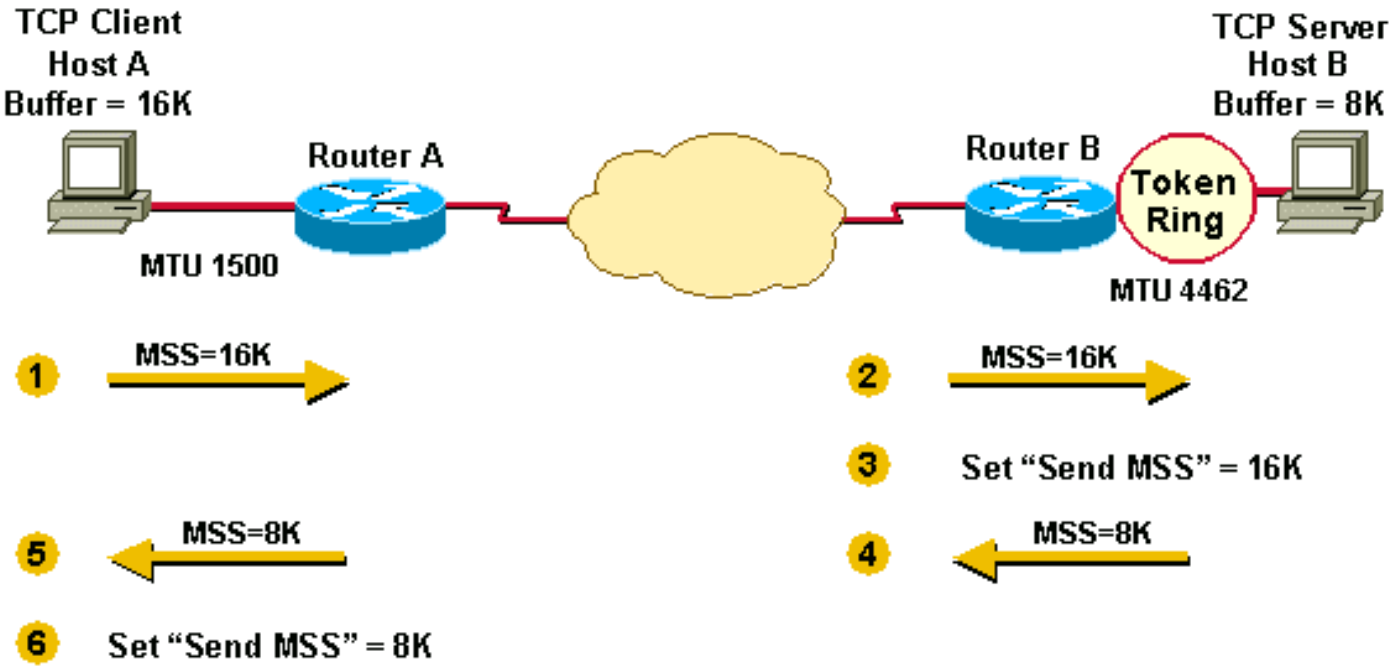
示例 1 说明了首次实施 MSS 的方式。

主机 A 的缓冲区为 16K，主机 B 的缓冲区为 8K。它们发送和接收各自的 MSS 值，并调整用于相互发送数据的发送 MSS。

主机 A 和主机 B 必须对大于接口 MTU 但小于发送 MSS 的 IPv4 数据报进行分段，因为 TCP 堆栈将 16K 或 8K 字节的数据从堆栈向下传递到 IPv4。

对于主机 B，数据包被分段先进入令牌环 LAN，然后再进入以太网 LAN。

示例 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。

为了帮助在 TCP 连接终端避免 IPv4 分段，MSS 值的选择被改为最小缓冲区大小和传出接口的 MTU (- 40 字节)。

由于 MSS（TCP 数据大小）不包括 20 字节的 IPv4 报头和 20 字节的 TCP 报头，因此 MSS 值比 MTU 值要小 40 个字节。

MSS 基于默认报头大小；发送方堆栈必须减去IPv4报头的相应值，而TCP报头取决于使用的TCP或IPv4选项。

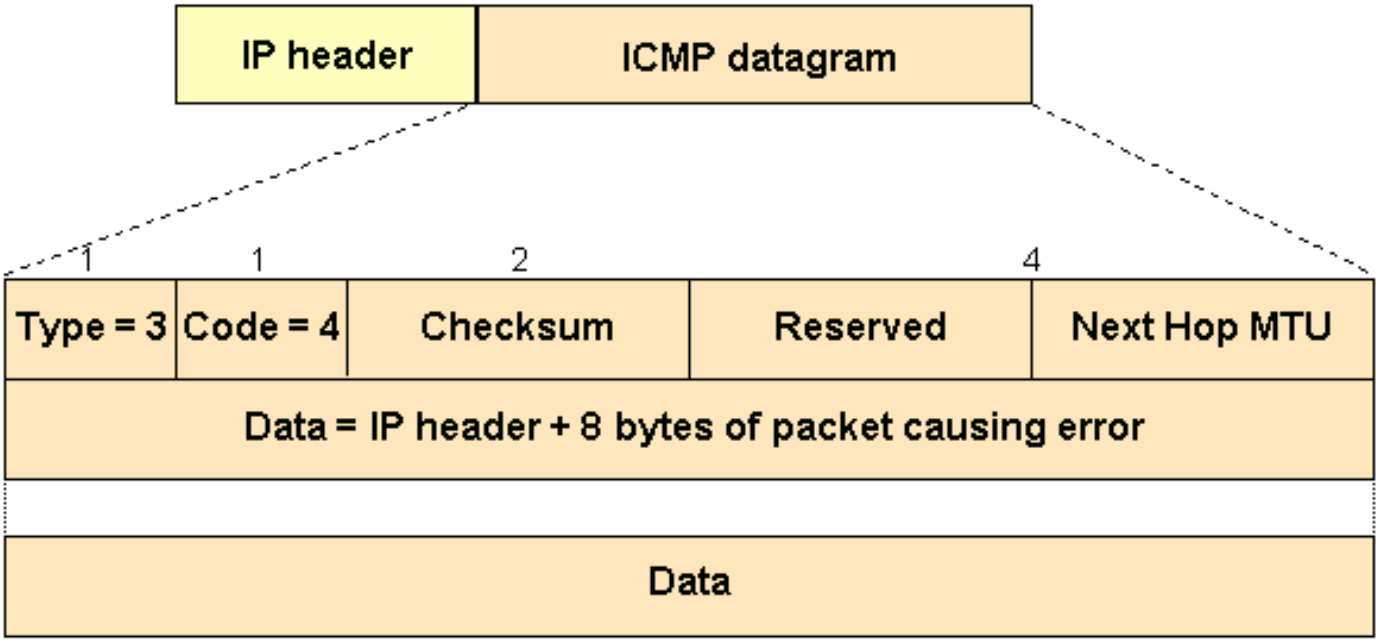
MSS 目前的工作方式是，每个主机首先将其传出接口 MTU 与自己的缓冲区进行比较，然后选择最小的值作为发送 MSS 值。

主机然后比较收到的 MSS 大小和自己的接口 MTU，然后选择两者之间的较小值。

示例 2 说明了发送端为避免在本地和远程线路上分段而需要额外采取的步骤。

每个主机在相互发送自己的 MSS 值之前都会考虑传出接口的 MTU。这有助于避免分段。

示例 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 接收来自主机 A 的发送 MSS (1460)，并将其与自己的出站接口 MTU 的值 - 40 (4422) 进行比较。
3. 主机 B 将较低的值 (1460) 设置为 MSS，以便向主机 A 发送 IPv4 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 接收来自主机 B 的发送 MSS (4422)，并将其与自己的出站接口 MTU 的值 - 40 (1460) 进行比较。
6. 主机 A 将较低的值 (1460) 设置为 MSS，以便向主机 B 发送 IPv4 数据报。

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

在示例 2 中，由于主机考虑到各自的传出接口 MTU，因此 TCP 连接终端不会发生分段。

如果数据包遇到 MTU 低于任一主机出站接口的 MTU 的链路，则数据包在路由器 A 和路由器 B 之间的网络中仍会进行分段。

PMTUD 是什么？

TCP MSS 负责处理 TCP 连接两个终端上的分段，但它不处理在这两个终端中间存在较小 MTU 链路的情况。

为了避免在终端之间的路径上出现分段，开发了 PMTUD。它用于动态确定从数据包源到数据包目的地路径上的最小 MTU。

注意：PMTUD 仅受 TCP 和 UDP 支持。其他协议不支持 PMTUD。如果在某个主机上启用 PMTUD，则来自该主机的所有 TCP 和 UDP 数据包都将设置 DF 位。

如果主机发送设置了 DF 位的完整 MSS 数据包，则在收到数据包需要分段的信息时，PMTUD 将减少连接的发送 MSS 值。

主机会记录目的地的 MTU 值，因为它在其路由表中使用此 MTU 值创建主机 (/32) 条目。

如果某路由器尝试将设置了 DF 位的 IPv4 数据报转发到 MTU 小于数据包大小的链路上，该路由器会丢弃此数据包并向 IPv4 数据报源返回“Destination Unreachable”（目的地不可达）的互联网控制消息协议 (ICMP) 消息，同时还会返回一个代码，指示“需要分段但设置了 DF”（类型 3，代码 4）。

当源站收到该 ICMP 消息时，会降低发送 MSS；当 TCP 重新传输该分段时，源站将使用较小的分段大小。

以下示例说明了在启用 命令后在路由器上显示的“fragmentation needed and DF set”（需要分段并设置 DF）ICMP 消息：**debug ip icmp**

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

下图显示了“需要分段和设置 DF”、“无法到达目标”消息的 ICMP 报信头的格式。

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

根据 RFC 1191，如果某路由器返回 ICMP 消息“需要分段但 DF 已设置”，则该路由器必须在根据 ICMP 规范 RFC 792 标记为“未使用”的低阶 16 位 ICMP 附加报头字段中，使用下一跳网络的 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。

为此，RFC 1191 还提供一个表，其中列出了在 PMTUD 期间 MTU 应当减少的建议值。

通过使用这些值，主机可以更快地获取合理的发送 MSS 值，如以上示例所示。

由于发送端和接收端之间的路径会动态变化，因此会对所有数据包持续执行 PMTUD。

每次发送端收到“Cannot Fragment”（无法分段）ICMP 消息时，它都会更新路由信息，PMTUD 即存储于该路由信息中。

执行 PMTUD 时，可能会发生以下两种情况：

1. 数据包可能会直接到达接收端，而不会被分段。

注意：路由器为了保护 CPU 免受 DoS 攻击，会将其发送的 ICMP 不可达消息的数量限制为每秒 2 个。因此，在这种情况下，如果您拥有这样的网络场景：您希望路由器每秒需要响应两条以上的 ICMP 消息（类型 = 3，代码 = 4）（可以是不同的主机），则需要使用 命令禁用对 ICMP 消息的限制。**no ip icmp rate-limit unreachable [df] interface**

---

2. 发送端从通往接收端的路径中的各跳收到“Cannot Fragment”（无法分段）ICMP 消息。

PMTUD 在 TCP 流量的两个方向上独立执行。

在某些情况下，数据流一个方向上的 PMTUD 会触发其中一个终端站降低发送 MSS，而另一个终端站则保持原来的发送 MSS，因为它发送的 IPv4 数据报从未大到足以触发 PMTUD。

示例3中描述的HTTP连接就是一个示例。TCP客户端发送小数据包，服务器发送大数据包。

在此例中，只有服务器发送的大数据包（大于 576 字节）才会触发 PMTUD。

客户端发送的数据包较小（小于 576 字节），不会触发 PMTUD，因为它们不需要进行分段来通过 576 MTU 链路。

示例 3



示例 4 显示的是非对称路由示例，其中一个路径的最小 MTU 比另一个路径的小。

采用不同的路径在两个终端之间发送和接收数据时，将会出现非对称路由。

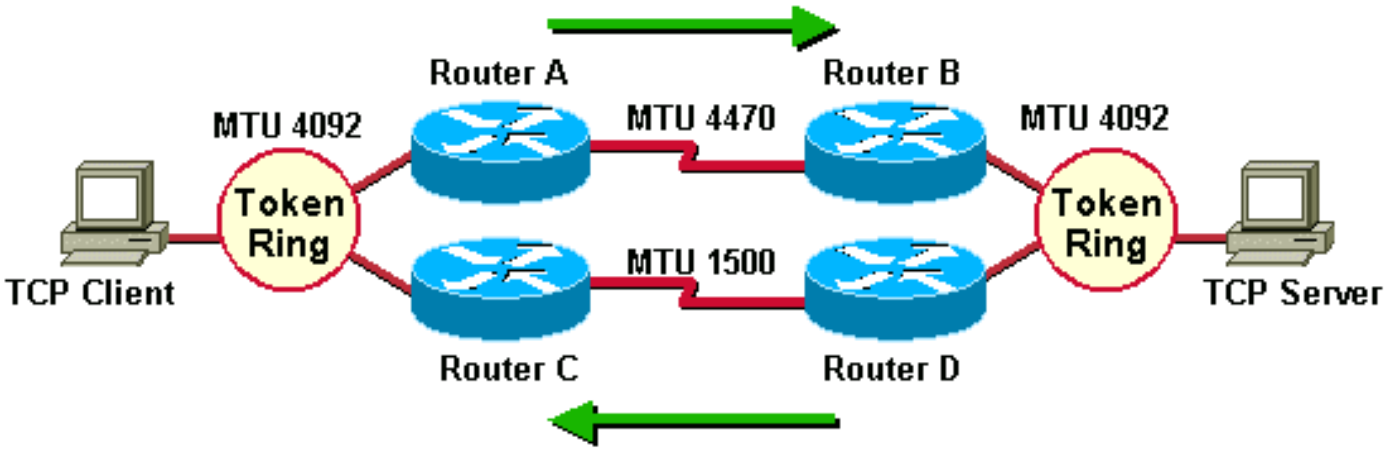
在此示例中，PMTUD 仅在 TCP 流的一个方向上触发降低发送 MSS。

从 TCP 客户端到服务器的流量会经过路由器 A 和路由器 B，而从服务器到客户端的返回流量会经过路由器 D 和路由器 C。

当 TCP 服务器向客户端发送数据包时，PMTUD 将触发服务器降低发送 MSS，因为路由器 D 必须在向路由器 C 发送 4092 字节数据包之前，对这些数据包进行分段。

相反，客户端永远不会收到“Destination Unreachable”（目的地不可达）ICMP 消息以及指示“需要分段但设置了 DF”的代码，因为路由器 A 在将数据包通过路由器 B 发送到服务器时不必对数据包分段。

示例 4



注意：可以使用 ip tcp path-mtu-discovery 命令对路由器发起的 TCP 连接（例如 BGP 和 Telnet）启用 TCP MTU 路径发现。

PMTUD 相关问题

以下是可能破坏 PMTUD 的事项。

- 路由器丢弃数据包，但不发送 ICMP 消息。（不常见）
- 路由器生成并发送 ICMP 消息，但路由器或此路由器与发送端之间的防火墙阻止该 ICMP 消息。（常见）
- 路由器生成并发送 ICMP 消息，但发送端忽略该消息。（不常见）

以上三项中的第一个和第三个问题通常是错误导致的结果，但中间的一项是常见问题。

实施 ICMP 数据包过滤器时，往往会阻止所有的 ICMP 消息类型，而非仅阻止特定的 ICMP 消息类型。

除“不可达”或“超时”类型的消息外，数据包过滤器可以阻止所有的 ICMP 消息类型。

PMTUD 的成功或失败取决于传送到 TCP/IPv4 数据包发送端的 ICMP 不可达消息。

ICMP 超时消息对于其他的 IPv4 问题非常重要。

下面显示了在路由器上实现的此类数据包过滤器示例。

```
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
```

还可以采用其他技术来缓解完全阻止 ICMP 的问题。

- 清除路由器上的 DF 位并允许分段。（不过，这不是一个好主意。有关详细信息，请参阅 IP 分段问题）。
- 使用接口命令 调整 TCP MSS 选项值 MSS。**ip tcp adjust-mss <500-1460>**

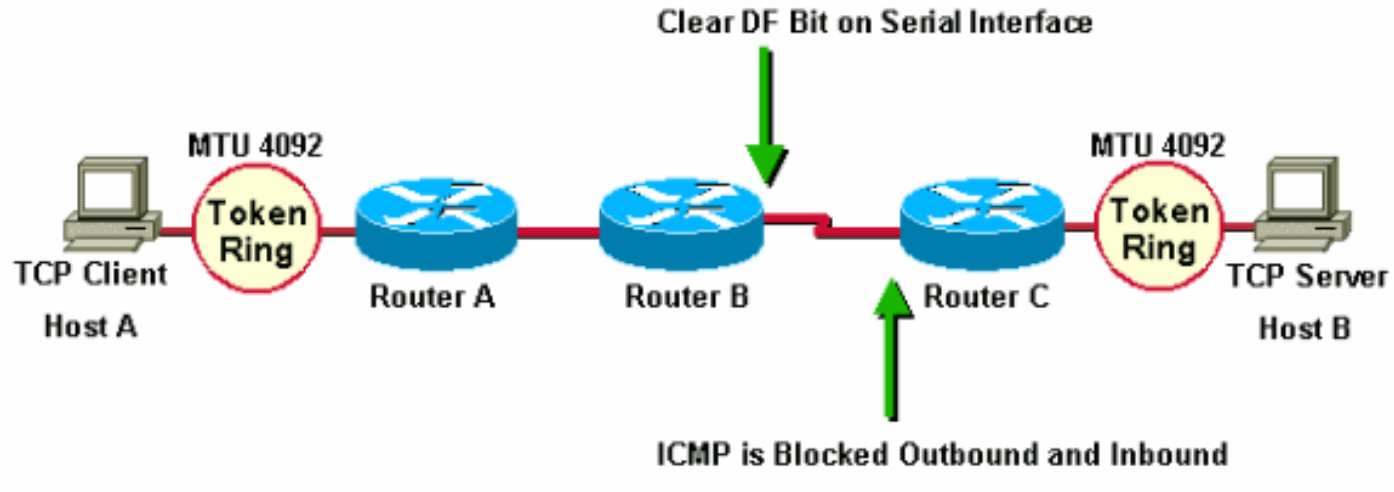
在下一个示例中，路由器 A 和路由器 B 位于同一管理域中。路由器 C 不可访问并且会阻止 ICMP，因此，PMTUD 将中断。

在这种情况下，解决方案是在路由器 B 的两个方向上清除 DF 位，从而允许分段。这可以通过策略路由来完成。

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
```



另一个选择是更改流经路由器的 SYN 数据包的 TCP MSS 选项值（在思科 IOS® 12.2(4)T 及更高版本中可用）。

这会减小 TCP SYN 数据包中的 MSS 选项值，使其小于 命令中的值 (1460)。**ip tcp adjust-mss**

这样，TCP 发送端发送的分段就不会超过该值。

IPv4 数据包将比 MSS 值（1460 字节）大 40 字节，即大小为 1500 字节，从而包含 TCP 报头（20 字节）和 IPv4 报头（20 字节）。

您可以使用 命令调整 TCP SYN 数据包的 MSS。**ip tcp adjust-mss**此语法会将 TCP 分段的 MSS 值减小到 1460 字节。

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
```

随着 IPv4 隧道部署越来越广泛，IPv4 分段问题也变得越来越普遍。

隧道会增加分段，因为隧道封装会增加数据包大小的“开销”。

例如，添加通用路径封装 (GRE) 会将 24 字节添加到数据包，增加之后，数据包需要进行分段，因为它大于出站 MTU。

需要 PMTUD 的常见网络拓扑

在中间链路 MTU 小于终端链路 MTU 的网络中，需要实施 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。两端的令牌环（或 FDDI）MTU 大于中间的以太网 MTU。
- 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)。

GRE、IPv4sec 和 L2TP 等隧道协议的报头和报尾也需要空间。这也会降低出站接口的有效 MTU。

隧道

隧道是思科路由器上的一个逻辑接口，用于在传输协议内封装乘客数据包。

这种架构旨在为点对点封装方案的实施提供服务。隧道接口具有以下三个主要组件：

- 乘客协议（AppleTalk、Banyan VINES、CLNS、DECnet、IPv4 或 IPX）
- 运载载波协议 - 以下封装协议之一：
    - GRE - 思科的多协议运载协议。有关详细信息，请参阅 RFC 2784 和 RFC 1701。
    - IPv4-in-IPv4 隧道；有关详细信息，请参阅 RFC 2003。
- 传输协议 - 用于传输经过封装的协议。

本部分中的数据包展示了以 GRE 为封装协议、以 IPv4 为传输协议的 IPv4 隧道概念。

乘客协议也是IPv4。在这种情况下，IPv4既是传输协议，也是乘客协议。

正常数据包

IPv4	TCP	Telnet

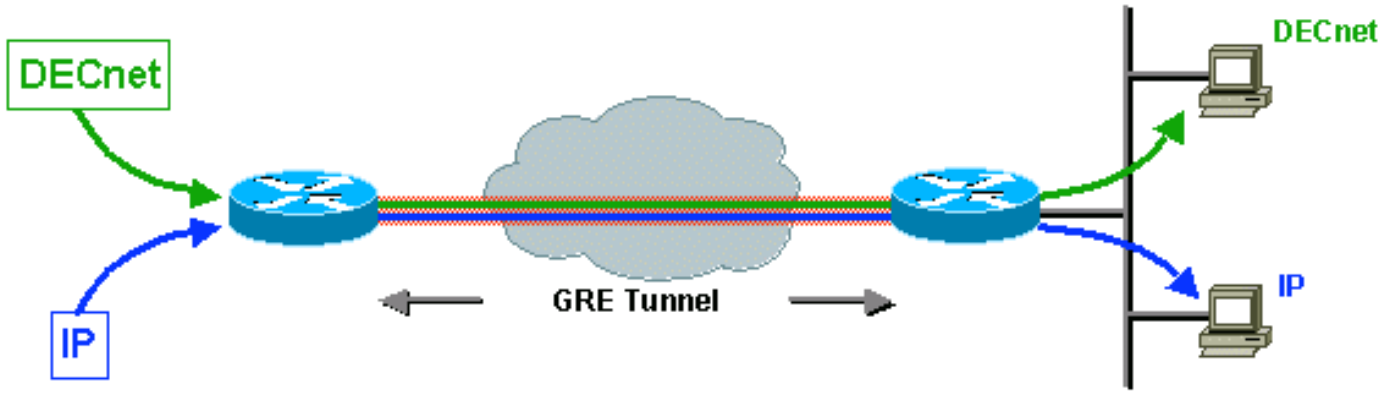
隧道数据包

IPv4	GRE	IPv4	TCP	Telnet

- IPv4 为传输协议。
- GRE 是封装协议。
- IPv4 为乘客协议。

在下一封装示例中，IPv4 和 DECnet 为乘客协议，GRE 为承载协议。

这说明了运载协议可以封装多个乘客协议，如下图所示。



在两个不连续的非 IPv4 网络被一个 IPv4 主干网隔开的情况下，网络管理员会考虑建立隧道。

如果不连续网络运行 DECnet，管理员可以选择通过在主干网中配置 DECnet 来连接（或不连接）这些网络。

管理员不希望允许 DECnet 路由使用主干网带宽，因为这可能会干扰 IPv4 网络的性能。

一种可行的备选方案就是在 IPv4 主干网上通过隧道传输 DECnet。隧道解决方案将 DECnet 数据包封装在 IPv4 内，并将其通过主干网发送到隧道终端；DECnet 数据包在隧道终端删除封装，然后通过 DECnet 路由到其目的地。

将流量封装在另一个协议内具有以下优点：

- 终端使用专用地址 (RFC 1918)，并且主干网不支持对这些地址进行路由。
- 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。

此后，IPv4 同时用作乘客协议和传输协议。

有关隧道接口的注意事项

以下是建立隧道时的注意事项。

- Cisco IOS® 版本 11.1 中引入了 GRE 隧道快速交换，版本 12.0 中引入了 CEF 交换。
- 多点 GRE 隧道的 CEF 交换功能是在版本 12.2(8)T 中引入的。
- 在早期版本的 Cisco IOS® 中，仅支持进程交换，因此隧道终端上的封装和解封操作速度很慢。
- 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。
- 如果通过防火墙实施隧道传输，则会绕过正在隧道传输的乘客协议。因此，为了对乘客协议执行策略，建议在隧道终端部署防火墙功能。
- 由于延迟增加，隧道传输会导致具有有限计时器的传输协议（例如 DECnet）出现问题。
- 如果隧道穿过具有不同速度链路（例如快速 FDDI 环）的环境以及通过缓慢的 9600 bps 电话线，会造成数据包重新排序问题。一些乘客协议在混合媒体网络中的运行效率非常低下。
- 点对点隧道会消耗物理链路上的带宽。在多个点对点隧道中，每个隧道接口都有一个带宽，并且运行隧道的物理接口也有带宽。例如，如果在 10 Mbps 链路上运行 100 条隧道，则需要将隧道带宽设置为 100 Kbps。隧道的默认带宽是 9 Kb。
- 与真实链路相比，路由协议更偏爱隧道，因为隧道存在一种假象，看起来像是一个路径成本最低的单跳链路，但它实际上包含多跳，因此成本比其他路径更高。正确配置路由协议可缓解这种情况。考虑在隧道接口上运行与在物理接口上运行的路由协议不同的路由协议。
- 通过配置通往隧道目的地的合适静态路由来避免递归路由问题。递归路由指通往隧道目的地的最佳路径就是通过隧道本身。这种情况会导致隧道接口上下跳动。当存在递归路由问题时会出现此错误。

```
%TUN-RECURDOWN Interface Tunnel 0
temporarily disabled due to recursive routing
```

在隧道终端作为 PMTUD 参与者的路由器

路由器作为隧道终端时，在 PMTUD 中起到两个不同的作用。

- 路由器的第一个作用就是充当主机数据包的转发端。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，然后采取相应操作。

- 路由器将原始 IPv4 数据包封装到隧道数据包后，开始发挥第二个作用。在此阶段，路由器的作用对于 PMTUD 和隧道 IPv4 数据包来说更像主机。

当路由器充当第一个角色（转发主机 IPv4 数据包的路由器）时，该角色在路由器将主机 IPv4 数据包封装在隧道数据包内之前发挥作用。

如果路由器充当主机数据包的转发器，它将完成以下操作：

- 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 DF 位未设置）、封装和发送数据包；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。

一般情况下，可以选择先封装再分段（发送两个封装分段）或者先分段再封装（发送两个已封装分段）。

此部分详细介绍了两个示例，它们展示了 PMTUD 与通过示例网络的数据包的交互。

第一个示例展示的是在（隧道源中的）路由器充当转发路由器的角色时，数据包发生的情况。

为处理 PMTUD，路由器需要检查原始数据包的 DF 位和数据包大小，然后采取相应操作。

本示例对隧道使用 GRE 封装。GRE 在封装之前先进行分段。

后面的示例显示在封装后进行分段的方案。

在示例 1 中，DF 位未设置 (DF = 0)，GRE 隧道 IPv4 MTU 为 1476 (1500-24) 字节。

示例 1

1. 转发路由器（位于隧道源侧）从发送主机收到一个 1500 字节的数据报，其中 DF 位已清除 (DF = 0)。

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。

IPv4	1480 字节 TCP + 数据

2. 由于在添加 GRE 开销（24 字节）后，数据包大于 IPv4 MTU，因此转发路由器将数据报拆分为两个分段，一个分段为 1476 字节（20 字节 IPv4 报头 + 1456 字节 IPv4 负载），另一个为 44 字节（20 字节 IPv4 报头 + 24 字节 IPv4 负载）。

添加 GRE 封装后，数据包不大于传出物理接口的 MTU。

IP_0	1456 字节 TCP + 数据
IP_1	24 字节数据

3. 转发路由器将 GRE 封装（包括 4 字节 GRE 报头和 20 字节 IPv4 报头）添加到原始 IPv4 数据报的各个分段。

这两个 IPv4 数据报的长度现在分别为 1500 字节和 68 字节，它们被视为单独的 IPv4 数据报，而不是分段。

IPv4	GRE	IP_0	1456 字节 TCP + 数据
IPv4	GRE	IP_1	24 字节数据

4. 隧道目标路由器从原始数据报的每个分段中去除 GRE 封装，从而留下两个长度为 1476 字节和 24 字节的 IPv4 分段。

路由器向接收主机单独转发这些 IPv4 数据报分段。

IP_0	1456 字节 TCP + 数据
IP_1	24 字节数据

5. 接收主机将这两个分段重组为原始数据报。

IPv4	1480 字节 TCP + 数据

示例 2 描绘了转发路由器在网络拓扑环境中的作用。

路由器仍充当转发路由器的角色，但这次设置了 DF 位 (DF = 1)。

示例 2

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

IPv4	1480 字节 TCP + 数据

2. 由于设置了 DF 位，并且数据报大小（1500 字节）大于 GRE 隧道 IPv4 MTU（1476 字节），因此路由器将丢弃该数据报，并向数据报源端发送“ICMP fragmentation needed but DF bit set”（需要分段但设置了 DF）ICMP 消息。

该 ICMP 消息提醒发送端 MTU 为 1476。

IPv4	ICMP MTU 1476

3. 发送主机接收 ICMP 消息；当它重新发送原始数据时，会使用 1476 字节的 IPv4 数据报。

IPv4	1456 字节 TCP + 数据

4. 现在此 IPv4 数据报的长度（1476 字节）数值上与 GRE 隧道的 IPv4 MTU 值相等，因此路由器将 GRE 封装添加到 IPv4 数据报中。

IPv4	GRE	IPv4	1456 字节 TCP + 数据

5. 接收路由器（位于隧道目标侧）删除 IPv4 数据报的 GRE 封装，然后将其发送到接收主机。

IPv4	1456 字节 TCP + 数据

以上就是路由器充当第二个角色（即作为 PMTUD 和隧道 IPv4 数据包的发送主机）时发生的情况。

该角色在路由器将原始 IPv4 数据包封装在隧道数据包内后发挥作用。

注意：默认情况下，路由器不会对其生成的GRE隧道数据包执行PMTUD。可以使用 命令对 GRE-IPv4 隧道数据包启用 PMTUD。**tunnel path-mtu-discovery**

示例 3 展示的是当主机发送的 IPv4 数据报小到足以适应 GRE 隧道接口上的 IPv4 MTU 时发生的情况。

在此情况下，可以设置或清除 DF 位（1 或 0）。

GRE 隧道接口未配置 命令，因此，路由器不会对 GRE-IPv4 数据包执行 PMTUD。**tunnel path-mtu-discovery**

示例 3

1. 位于隧道源侧的转发路由器从发送主机收到一个 1476 字节的数据报。

IPv4	1456 字节 TCP + 数据

2. 此路由器将 1476 字节的 IPv4 数据报封装到 GRE 内部，得到 1500 字节的 GRE IPv4 数据报。

GRE IPv4 报头中的 DF 位被清除 (DF = 0)。 然后，此路由器将此数据包转发到隧道目标。

IPv4	GRE	IPv4	1456 字节 TCP + 数据

3. 假设隧道源端和目的地端之间有一个路由器，存在一条 MTU 为 1400 的链路。

由于清除了 DF 位 (DF = 0)，因此该路由器会对隧道数据包进行分段。

请注意，此示例对最外层的 IPv4 进行分段，因此 GRE、内部 IPv4 和 TCP 报头将仅在第一个分段中显示。

IP_0	GRE	IP	1352 字节 TCP + 数据
IP_1	104 字节数据		

4. 隧道目的地路由器必须重组 GRE 隧道数据包。

IP	GRE	IP	1456 字节 TCP + 数据

5. 在重组 GRE 隧道数据包后，此路由器删除 GRE IPv4 报头，并继续发送原始 IPv4 数据报。

IPv4	1456 字节 TCP + 数据

示例 4 展示的是当路由器充当 PMTUD 和隧道 IPv4 数据包的发送主机这一角色时发生的情况。

这次，在原始 IPv4 报头中设置了 DF 位 (DF = 1)，并且配置了 命令，以便将 DF 位从内部 IPv4 报头复制到外部 (GRE + IPv4) 报头。**tunnel path-mtu-discovery**

示例 4

1. 位于隧道源侧的转发路由器从发送主机收到一个 1476 字节的数据报，其中 DF = 1。

IPv4	1456 字节 TCP + 数据

2. 此路由器将 1476 字节的 IPv4 数据报封装到 GRE 内部，得到 1500 字节的 GRE IPv4 数据报。

由于原始 IPv4 数据报设置了 DF 位，因此该 GRE IPv4 数据报也设置了 DF 位 (DF =1)。

然后，此路由器将此数据包转发到隧道目标。

IPv4	GRE	IPv4	1456 字节 TCP

3. 同样，假设在隧道源端和目的地端之间有一个路由器，存在一条 MTU 为 1400 的链路。

由于设置了 DF 位 (DF = 1)，因此该路由器不会对隧道数据包进行分段。

该路由器必须丢弃数据包，并向隧道源路由器（因为这是数据包的源 IPv4 地址）发送 ICMP 错误消息。

IPv4	ICMP MTU 1400

4. 位于隧道源侧的转发路由器收到此 ICMP 错误消息，并将 GRE 隧道的 IPv4 MTU 降低至 1376 (1400 - 24) 字节。

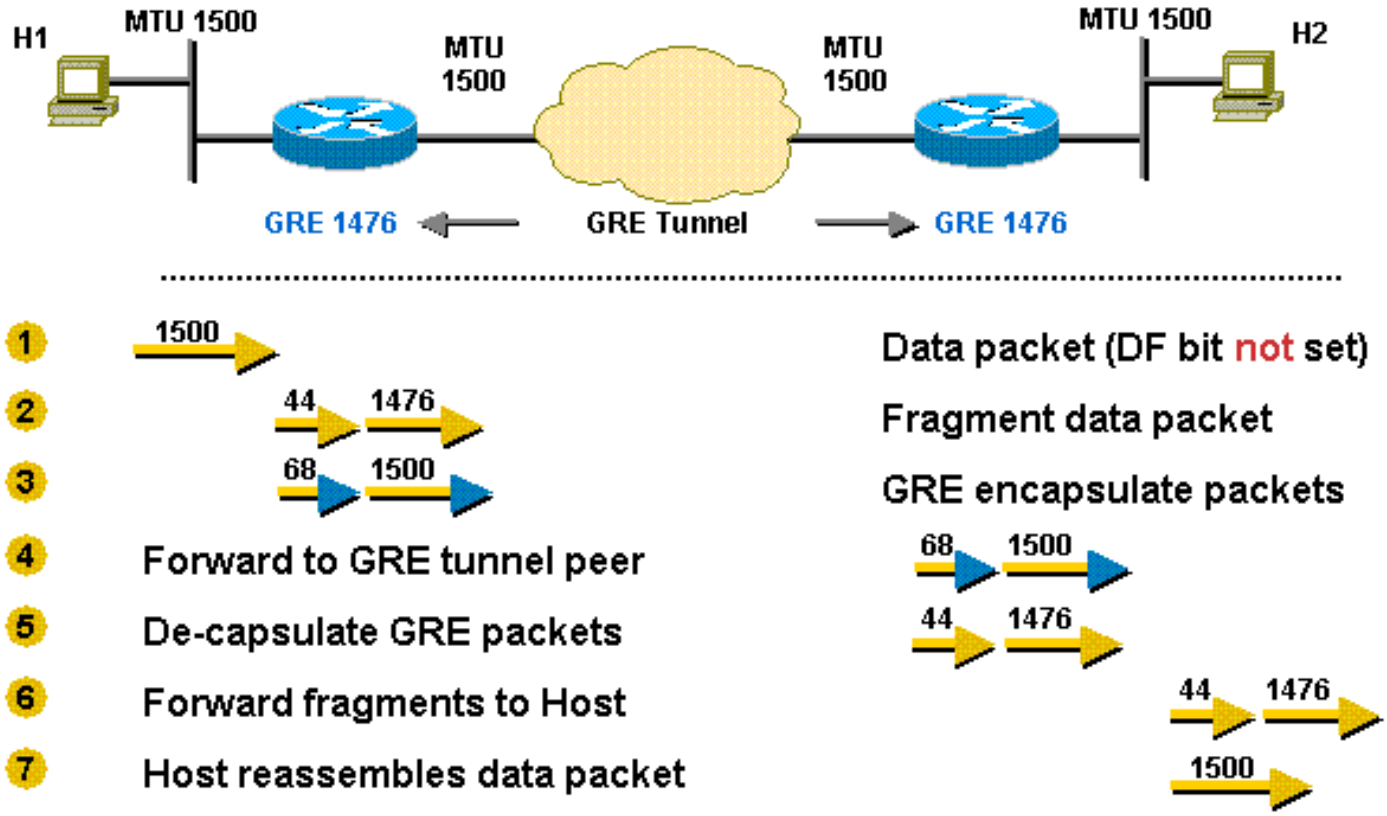
下次发送主机以 1476 字节的 IPv4 数据包重新传输数据，该数据包过大，因此该路由器会向发送端发送 ICMP 错误消息，其中指示 MTU 值为 1376。

当发送主机重新传输数据时，将以 1376 字节的 IPv4 数据包发送，此数据包将通过 GRE 隧道成功到达接收主机。

示例 5

此示例说明了 GRE 分段。在 GRE 封装之前需要对数据包进行分段，然后对数据包执行 PMTUD；当对 IPv4 数据包执行 GRE 封装时，不会复制 DF 位。

未设置 DF 位。GRE 隧道接口的 IPv4 MTU 默认比物理接口的 IPv4 MTU 少 24 字节，因此 GRE 接口的 IPv4 MTU 为 1476 字节，如下图所示。



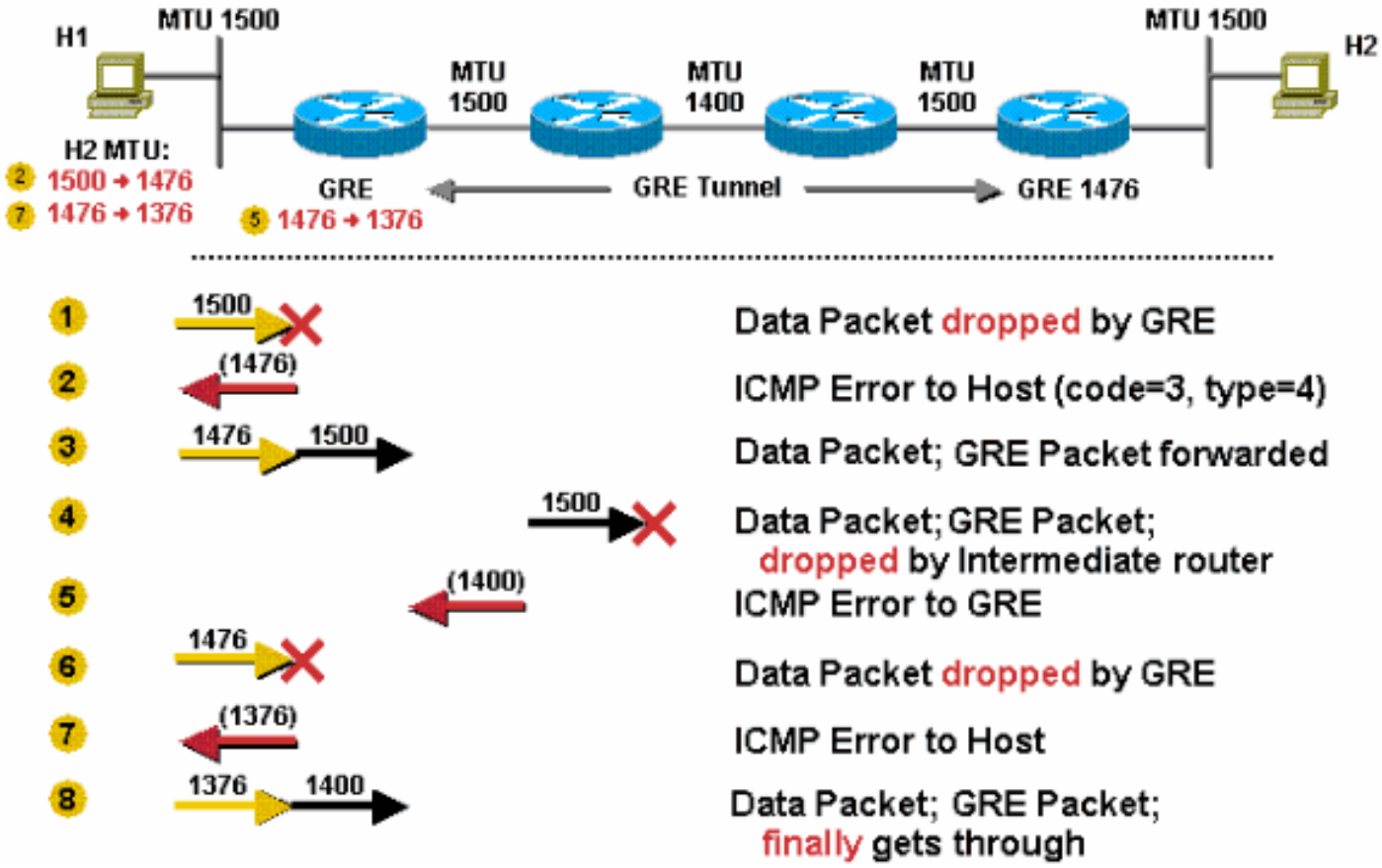
1. 发送端发送一个 1500 字节的数据包（20 字节 IPv4 报头 + 1480 字节 TCP 负载）。
2. 由于 GRE 隧道的 MTU 为 1476 字节，因此 1500 字节的数据包被分为两个 IPv4 分段（一个 1476 字节，一个 44 字节），每个分段预计加上另外 24 字节的 GRE 报头。
3. 每个 IPv4 分段增加 24 字节的 GRE 报头。现在，两个分段分别为 1500 字节 (1476 + 24) 和 68 字节 (44 + 24)。
4. 包含两个 IPv4 分段的 GRE + IPv4 数据包被转发到 GRE 隧道对等路由器。
5. GRE 隧道对等路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 IPv4 分段重组为原始 IPv4 数据报。

示例 6

此示例与示例 5 类似，但这次设置了 DF 位。通过 命令将路由器配置为对 GRE + IPv4 隧道数据包执行 PMTUD，并将 DF 位从原始 IPv4 报头复制到 GRE IPv4 报头。**tunnel path-mtu-discovery**

如果路由器收到面向 GRE + IPv4 数据包的 ICMP 错误，则会降低 GRE 隧道接口的 IPv4 MTU 值。

GRE 隧道的 IPv4 MTU 默认设置为比物理接口的 MTU 小 24 字节，因此此处的 GRE IPv4 MTU 为 1476。GRE 隧道路径中有一条 MTU 为 1400 字节的链路，如下图所示。



1. 路由器收到 1500 字节的数据包（20 字节 IPv4 报头 + 1480 字节 TCP 负载），然后丢弃该数据包。路由器丢弃数据包是因为该数据包大于 GRE 隧道接口上的 IPv4 MTU (1476)。
2. 路由器向发送端发送 ICMP 错误消息，告知发送端下一跳 MTU 为 1476 字节。主机将此信息记录在自己的路由表中，通常记录为通往目的地的主机路由。
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. 1500 字节的数据包不能穿越 1400 字节的链路，因此被中间路由器丢弃。
5. 中间路由器将向 GRE 路由器发送一个含有下一跳 MTU 为 1400 的 ICMP（类型 = 3，代码 = 4）。GRE 路由器将其降低至 1376 (1400 - 24) 字节，并在 GRE 接口上设置内部 IPv4 MTU 值。只有在您使用命令时才能看到此更**debug tunnel** 改；在命令的输出中看不到**show ip interface tunnel<#>**它。
6. 主机下次重新发送 1476 字节的数据包时，GRE 路由器将丢弃该数据包，因为它大于 GRE 隧道接口上当前的 IPv4 MTU (1376)。
7. GRE 路由器向发送端发送另一条 ICMP 消息（类型 = 3，代码 = 4），其中指示下一跳的 MTU 为 1376 字节，主机将用新值更新其当前信息。
8. 主机再次重新发送数据，但现在以较小的 1376 字节数据包发送；GRE 添加 24 字节的封装，然后转发。这次，数据包发送到 GRE 隧道对等体，在此，数据包进行解封并发送到目的主机。

注意：如果此场景中的转发路由器上未配置此命令，并且通过GRE隧道转发的数据包中设置了DF位，则主机1仍能成功将TCP/IPv4数据包发送到主机2，但会在1400 MTU链路的中途分段**tunnel path-mtu-discovery**。此外，GRE 隧道对等体必须在解封装和转发数据包前对数据包进行重组。

纯 IPsec 隧道模式

IPv4 安全 (IPv4sec) 协议是一种基于标准的方法，用于确保 IPv4 网络上传输的信息保密、完整且真实。

IPv4sec 提供 IPv4 网络层加密。IPv4sec 通过添加至少一个 IPv4 报头（隧道模式）而加长 IPv4 数据包。

添加的报头的长度因 IPv4sec 配置模式而异，但每个数据包不会超过约 58 字节（封装安全负载 [ESP] 和 ESP 身份验证 [ESPauth]）。

IPv4sec 有两种模式：隧道模式和传送模式。

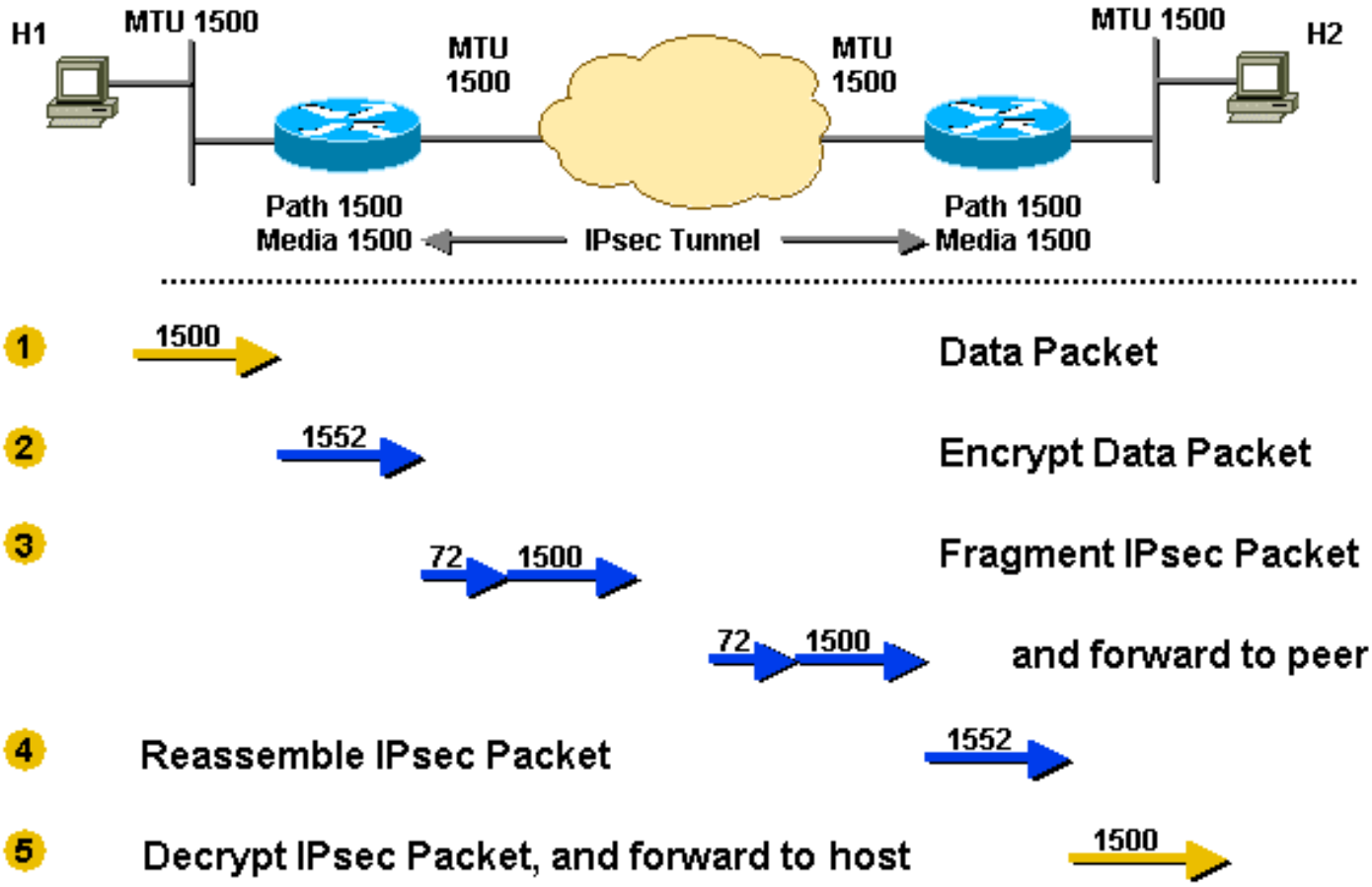
1. 隧道模式是默认模式。在隧道模式下，整个原始 IPv4 数据包都受到保护（加密、身份验证或两者同时执行），并由 IPv4sec 报头和报尾进行封装。然后，新的 IPv4 报信头附加到数据包前面，用于将 IPv4sec 终端（对等体）指定为源端和目标端。隧道模式可用于任何单播 IPv4 流量，但如果将 IPv4sec 用于保护来自 IPv4sec 对等体背后的主机流量，则必须使用隧道模式。例如，隧道模式可与虚拟专用网络 (VPN) 结合使用，其中通过一组 IPv4sec 对等体，一个受保护网络上的主机可向另一个受保护网络上的主机发送数据包。在 VPN 网络中，IPv4sec 隧道通过加密 IPv4sec 对等路由器间的流量来保护主机之间的 IPv4 流量。
2. 在传输模式（在转换定义中使用子命令 配置）下，只有原始 IPv4 数据包的负载受到保护（加密和/或身份验证）。**mode transport** 负载由 IPv4sec 报头和报尾封装。原始 IPv4 报头保持不变，只不过 IPv4 协议字段更改为 ESP (50)，而原始协议值保存在 IPv4sec 报尾，并在解密数据包时恢复。只有在要保护的 IPv4 流量位于 IPv4sec 对等体之间，并且数据包上的源和目标 IPv4 地址都与 IPv4sec 对等体地址相同时，才使用传送模式。正常情况下，只有在使用另一种隧道协议（例如 GRE）来首先封装 IPv4 数据包，然后使用 IPv4sec 保护 GRE 隧道数据包的情况下，才会使用 IPv4sec 传送模式。

对于数据包和自己的数据包，IPv4sec 始终会执行 PMTUD。可以使用 IPv4sec 配置命令来修改对 IPv4sec IPv4 数据包的 PMTUD 处理，IPv4sec 可以清除、设置 DF 位，或将 DF 位从数据包 IPv4 报头复制到 IPv4sec IPv4 报头。该功能称为“DF 位覆盖功能”。

注意：使用IPv4sec进行硬件加密时，请避免封装后的分段。硬件加密会导致约 50 Mbs 的吞吐量（取决于硬件），但如果对 IPv4sec 数据包进行分段，则会丢失 50% 至 90% 的吞吐量。这是因为分段的 IPv4sec 数据包重组时会经过进程交换，然后传送给硬件加密引擎进行解密。吞吐量的丢失会导致硬件加密吞吐量下降至软件加密的性能级别 (2-10 Mbs)。

示例 7

本场景描述的是 IPv4sec 分段过程。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. 路由器收到发往主机 2 的 1500 字节的数据包（20 字节 IPv4 报头 + 1480 字节 TCP 负载）。
2. 1500 字节的数据包经过 IPv4sec 加密，增加了 52 字节的开销（IPv4sec 报头、报尾和另外的 IPv4 报头）。 现在 IPv4sec 需要发送 1552 字节的数据包。由于出站接口的 MTU 为 1500 字节，因此必须对此数据包进行分段。
3. IPv4sec 数据包被拆分为两个分段。在分段过程中，会给第二个分段另外添加 20 字节的 IPv4 报头，从而产生一个 1500 字节的分段和一个 72 字节的 IPv4 分段。
4. IPv4sec 隧道对等路由器接收分段，剥离附加的 IPv4 报头，并将 IPv4 分段合并为原始 IPv4sec 数据包。然后 IPv4sec 解密该数据包。
5. 最后，路由器将 1500 字节的原始数据包转发到主机 2。

示例 8

此示例与示例 6 类似，区别在于在本例中，原始数据包中设置了 DF 位，并且在 IPv4sec 隧道对等体之间存在一条链路，该链路的 MTU 低于其他链路的 MTU。

此示例演示 IPv4sec 对等路由器如何充当两种 PMTUD 角色（如在隧道终端作为 PMTUD 参与者的路由器部分所述）。

IPv4sec PMTU 由于分段需要而改为更低的值。

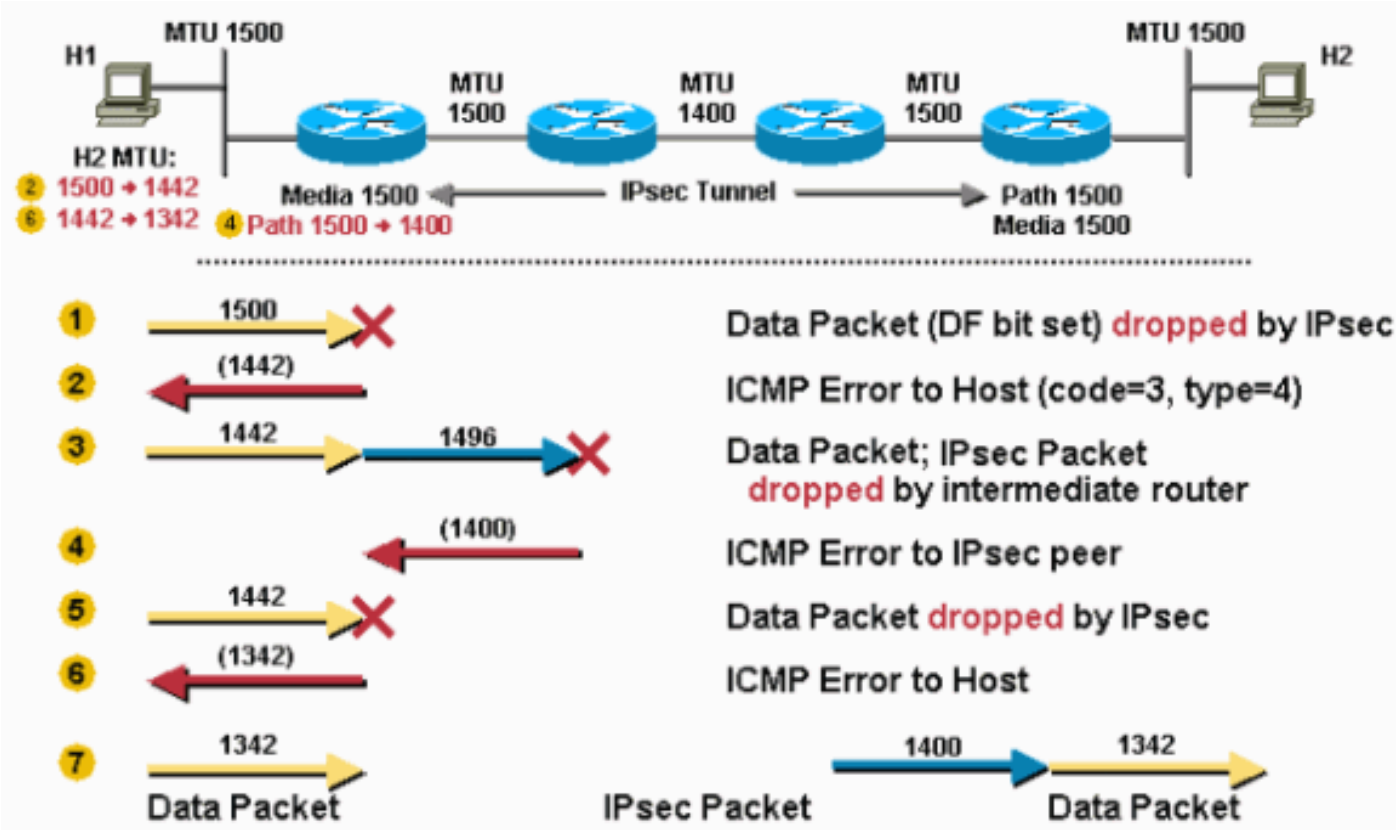
当 IPv4sec 加密数据包时，会将 DF 位从内部 IPv4 报头复制到外部 IPv4 报头。

介质 MTU 和 PMTU 值存储在 IPv4sec 安全关联 (SA) 中。

介质 MTU 基于出站路由器接口的 MTU，而 PMTU 基于 IPv4sec 对等体间路径上看到的最小

MTU。

IPv4sec 在尝试对数据包进行分段之前会封装/加密数据包，如下图所示。



1. 路由器收到 1500 字节的数据包，然后丢弃该数据包，因为加上 IPv4sec 开销后，数据包大于 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)。 此 58 字节为使用 IPv4sec ESP 和 ESPauth 时的最大 IPv4sec 开销。实际的 IPv4sec 开销可能比该值少 7 个字节。主机 1 将此信息记录在自己的路由表中，通常记录为通往目的地（主机 2）的主机路由。
3. 主机1将主机2的PMTU降低到1442，因此主机1在重新向主机2传输数据时发送较小（1442字节）的数据包。路由器接收1442字节的数据包，而IPv4sec增加52字节的加密开销，因此产生的IPv4sec数据包为1496字节。由于此数据包在报头中已设置 DF 位，因此会被链路 MTU 为 1400 字节的中间路由器丢弃。
4. 丢弃数据包的中间路由器向 IPv4sec 数据包的发送端（第一个路由器）发送一条 ICMP 消息，告知发送端下一跳 MTU 为 1400 字节。这个值记录在 IPv4sec SA PMTU 中。
5. 主机 1 下次重新传输 1442 字节的数据包时（主机未收到面向该数据包的确认消息），IPv4sec 将丢弃该数据包。路由器会丢弃该数据包，因为 IPv4sec 开销添加到数据包后会使数据包大于 PMTU (1400)。
6. 路由器向主机 1 发送 ICMP 消息，告知主机现在下一跳 MTU 为 1342。(1400 - 58 = 1342)。主机 1 再次记录此信息。
7. 当主机 1 再次重新传输数据时，它会使用较小的数据包 (1342)。 此数据包不需要分段，并将通过 IPv4sec 隧道成功到达主机 2。

GRE 与 IPv4sec 协同工作

使用 IPv4sec 来加密 GRE 隧道时，分段和 PMTUD 的交互会更为复杂。

IPv4sec 和 GRE 以这种方式组合是因为 IPv4sec 不支持 IPv4 组播数据包，这意味着在 IPv4sec VPN 网络上无法运行动态路由协议。

GRE 隧道支持组播，因此可先使用 GRE 隧道加密 GRE IPv4 单播数据包中的动态路由协议组播数据包，然后再使用 IPv4sec 加密单播数据包。

在执行此操作时，通常是在 GRE 之上以传输模式部署 IPv4sec，因为 IPv4sec 对等体和 GRE 隧道终端（路由器）相同，而传输模式可节省 20 字节的 IPv4sec 开销。

有个有趣的案例，其中 IPv4 数据包被拆分为两个分段，并采用 GRE 封装。

在这种情况下，IPv4sec 将看到两个独立的 GRE + IPv4 数据包。通常在默认配置下，其中一个数据包会因为在加密后变得过大而需要被分段。

IPv4sec 对等体必须在解密之前重组此数据包。这种发送路由器上的“两次分段”（GRE 前一次，IPv4sec 后一次）会提高时延，并降低吞吐量。

重组采用进程交换，因此每次重组时接收路由器就会发生一次 CPU 命中。

如果考虑 GRE 和 IPv4sec 开销而将 GRE 隧道接口的“ip mtu”设置得足够低，则可以避免这种情况（默认情况下，GRE 隧道接口“ip mtu”被设置为实际传出接口 MTU 值 - GRE 开销字节数）。

下表列出了假定传出物理接口的 MTU 为 1500 时每个隧道/模式组合的建议 MTU 值。

隧道组合	需要的特定 MTU	建议的 MTU
GRE + IPv4sec（传送模式）	1440 字节	1400 字节
GRE + IPv4sec（隧道模式）	1420 字节	1400 字节


注意：建议将 MTU 值设置为 1400 字节 ，因为该值涵盖了最常见的 GRE + IPv4sec 模式组合。此外，允许额外的20或40字节开销也没有明显的缺点。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。

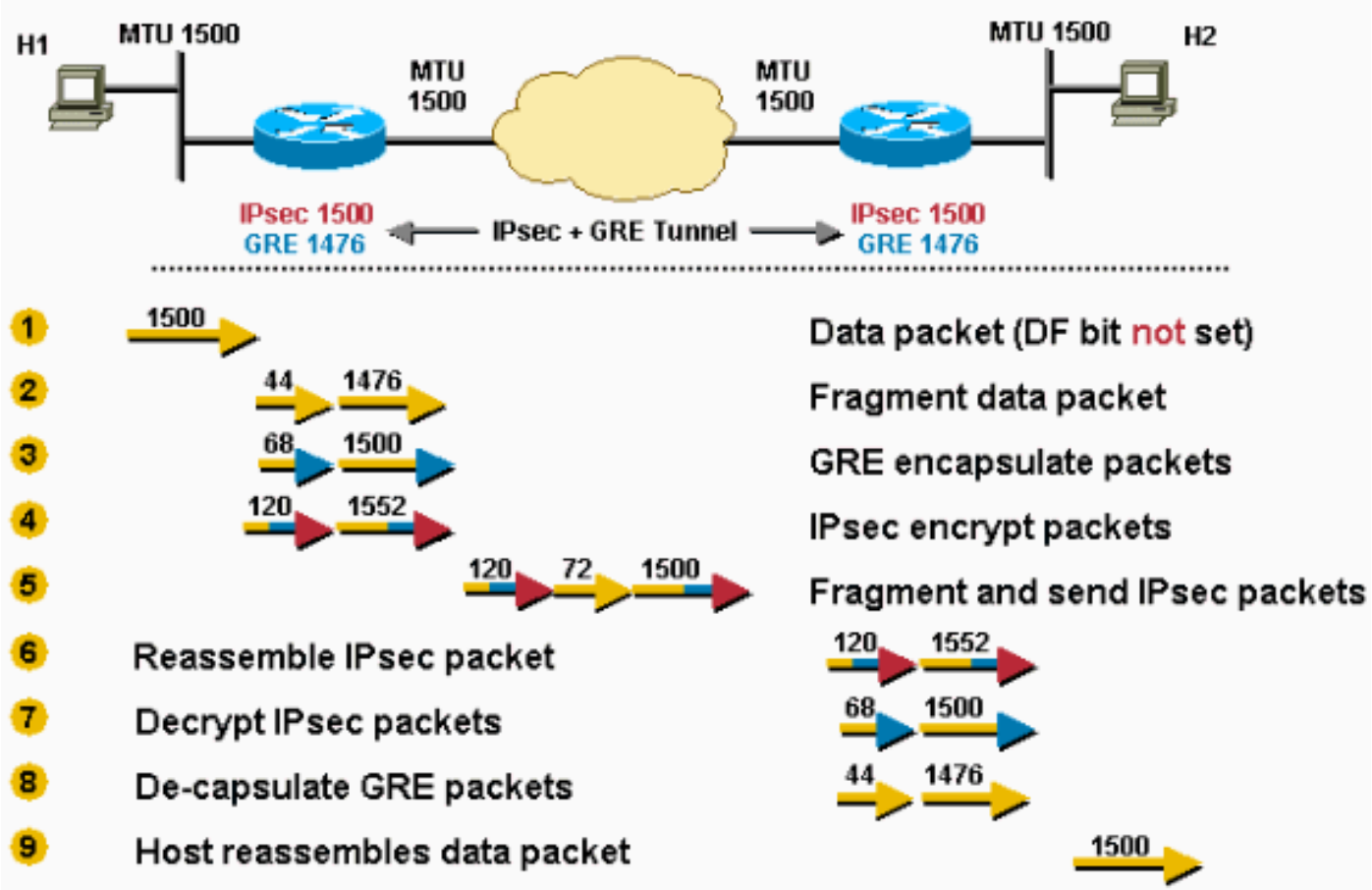
示例 9

IPv4sec 部署在 GRE 之上。传出物理 MTU 为 1500，IPv4sec PMTU 为 1500，GRE IPv4 MTU 为 1476 (1500 - 24 = 1476)。

因此，TCP/IPv4 数据包会进行两次分段，一次在 GRE 之前，一次在 IPv4sec 之后。

在 GRE 封装之前会对数据包进行分段，在 IPv4sec 加密之后会再对其中一个 GRE 数据包进行分段。

在本场景下，如果在 GRE 隧道上配置“ip mtu 1440”（IPv4sec 传送模式）或“ip mtu 1420”（IPv4sec 隧道模式），则可以消除两次分段的可能性。



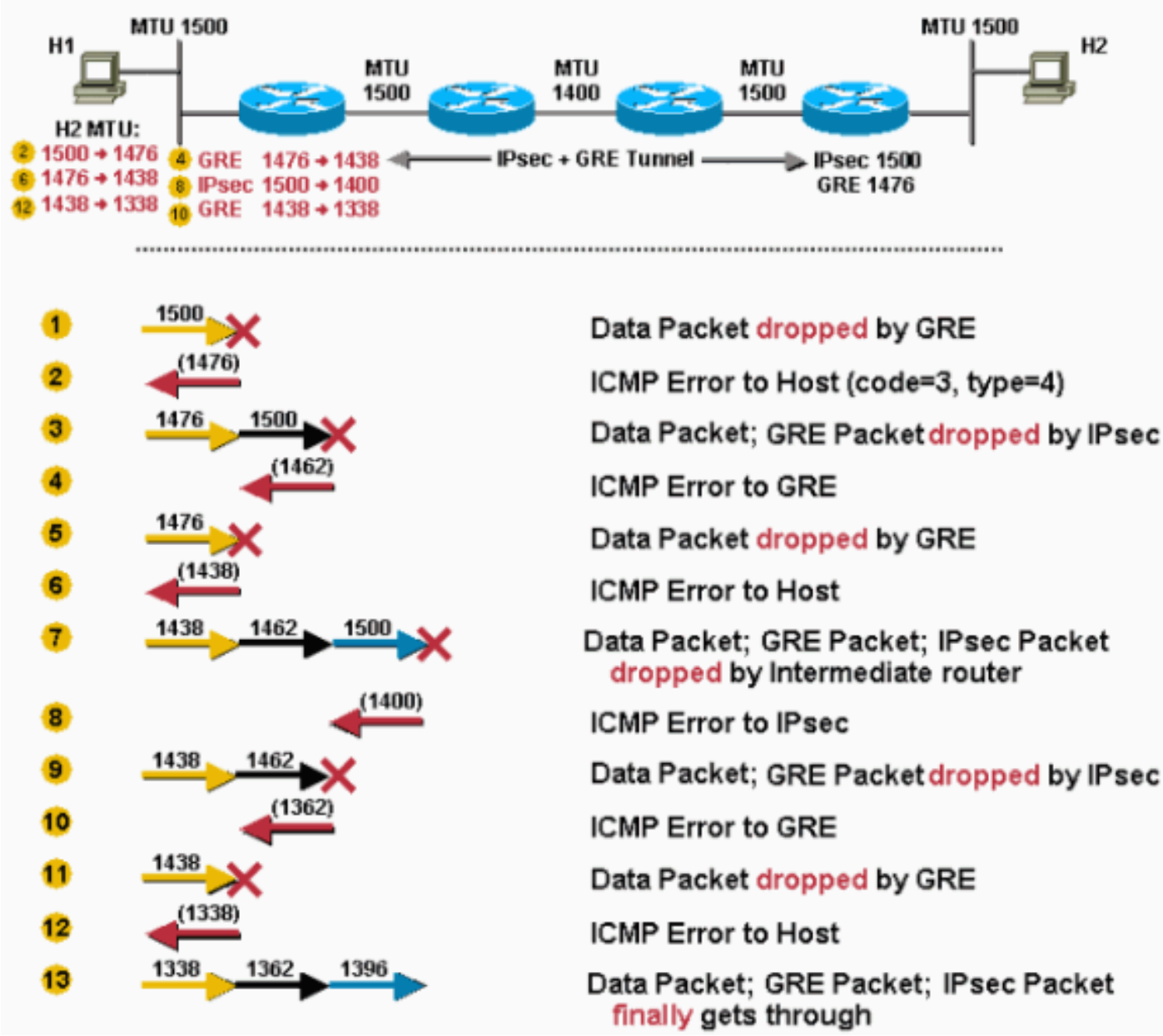
1. 路由器收到一个 1500 字节的数据报。
2. 在封装之前，GRE 将 1500 字节的数据包拆分为两个分段，一个 1476 字节 (1500 - 24 = 1476)，另一个 44 字节（24 字节数据 + 20 字节 IPv4 报头）。
3. GRE 封装 IPv4 分段，该过程将导致每个数据包增加 24 字节。因而将产生两个 GRE + IPv4sec 字段，一个 1500 字节 (1476 + 24 = 1500)，另一个 68 字节 (44 + 24 = 68)。
4. IPv4sec 对两个数据包进行加密，这会对每个数据包添加 52 字节（IPv4sec 隧道模式）的封装开销，从而生成一个 1552 字节的数据包和一个 120 字节的数据包。
5. 由于 1552 字节的 IPv4sec 数据包大于出站 MTU (1500)，因此，路由器会将其分段。 1552 字节的数据包被拆分为 1500 字节的数据包和 72 字节的数据包（52 字节负载加上为第二个分段附加的 20 字节 IPv4 报头）。 三个数据包（1500 字节、72 字节和 120 字节）被转发到 IPv4sec + GRE 对等设备。
6. 接收路由器重组两个 IPv4sec 分段（1500 字节和 72 字节），以便获取原始 1552 字节的 IPv4sec + GRE 数据包。对于 120 字节的 IPv4sec + GRE 数据包无需任何操作。
7. IPv4sec 对 1552 字节和 120 字节的 IPv4sec + GRE 数据包进行解密，以便获取 1500 字节和 68 字节的 GRE 数据包。
8. GRE 解封装 1500 字节和 68 字节的 GRE 数据包，以便获取 1476 字节和 44 字节的 IPv4 数据包分段。然后这些 IPv4 数据包分段被转发到目标主机。
9. 主机 2 重组这些 IPv4 分段，以便获取原始 1500 字节的 IPv4 数据报。

场景 10 类似于场景 8，区别在于，隧道路径中有一条 MTU 较低的链路。这是从主机1发送到主机2的第一个数据包的最坏情况。在此场景的最后一步之后，主机1为主机2设置正确的PMTU，并且所有操作都可用于主机1和主机2之间的TCP连接。主机1与其他主机之间的TCP流（可通过IPv4sec + GRE隧道访问）只需完成场景10的最后三个步骤。

在本场景中，GRE 隧道上配置了 命令，并且来自主机 1 的 TCP/IPv4 数据包上设置了 DF 位。

tunnel path-mtu-discovery

示例 10



1. 路由器收到一个 1500 字节的数据包。由于已设置 DF 位，并且在添加 GRE 开销（24 字节）后，数据包的大小超过出站接口的“ip mtu”值，因而 GRE 无法对数据包分段或转发数据包，因此丢弃该数据包。
2. 路由器向主机 1 发送 ICMP 消息，以便该主机知晓下一跳 MTU 为 1476 (1500 - 24 = 1476)。
3. 主机 1 将其面向主机 2 的 PMTU 改为 1476，并在重新传输数据包时，发送较小的数据包。GRE 封装数据包，并将 1500 字节的数据包传递给 IPv4sec。由于 GRE 从内部 IPv4 报头中复制了 DF 位（已设置），并且加上 IPv4sec 开销（最大 38 字节）后，数据包因太大而无法传出物理接口，因此 IPv4sec 丢弃该数据包。
4. IPv4sec 向 GRE 发送 ICMP 消息，指出下一跳的 MTU 为 1462 字节（因为加密和 IPv4 开销最多会添加 38 字节）。GRE 将值 1438 (1462 - 24) 记录为隧道接口上的“ip mtu”。

> 注意：此值更改存储在内部，在命令输出中无法看**show ip interface tunnel<#>**到。只有在使用 命令时才能看到此更改。**debug tunnel**

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. 路由器向主机 1 发送 ICMP 消息，指明下一跳 MTU 为 1438。
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。这次 GRE 接受该数据包，对其进行封装，并将其传递给 IPv4sec 进行加密。
8. IPv4sec 数据包被转发到中间路由器并被丢弃，因为该路由器出站接口 MTU 为 1400。
9. 中间路由器向 IPv4sec 发送 ICMP 消息，指明下一跳 MTU 为 1400。IPv4sec 将该值记录在关联 IPv4sec SA 的 PMTU 值中。
10. 当主机 1 重新传输 1438 字节的数据包时，GRE 封装该数据包，然后将其传递给 IPv4sec。IPv4sec 丢弃该数据包，因为其已将自己的 PMTU 改为 1400。
11. IPv4sec 向 GRE 发送 ICMP 错误消息，指明下一跳 MTU 为 1362，并且 GRE 在内部记录值 1338。
12. 当主机 1 重新传输原始数据包（因为主机未收到确认消息）时，GRE 丢弃该数据包。
13. 路由器向主机 1 发送 ICMP 消息，指明下一跳 MTU 为 1338（1362 - 24 字节）。 主机 1 针对主机 2 将其 PMTU 减小至 1338。
14. 主机 1 重新传输 1338 字节的数据包，这次数据包可以最终顺利到达主机 2。

更多建议

在隧道接口上配置 命令有助于相同路由器上配置的 GRE 和 IPv4sec 的交互。**tunnel path-mtu-discovery**

如果未配置 命令，则在 GRE IPv4 报头中 DF 位始终为清除状态。**tunnel path-mtu-discovery**

这样便可对 GRE IPv4 数据包进行分段，即使已封装的数据 IPv4 报头设置了 DF 位（通常不允许对数据包进行分段）也是如此。

如果在 GRE 隧道接口上配置了 命令：**tunnel path-mtu-discovery**

1. GRE 将从数据 IPv4 报头中复制 DF 位到 GRE IPv4 报头。
2. 如果 GRE IPv4 报头中设置了 DF 位，且数据包在经过 IPv4sec 加密后，对于物理传出接口的 IPv4 MTU 而言“过大”，则 IPv4sec 将丢弃该数据包，并通知 GRE 隧道减小其 IPv4 MTU 的大小。
3. IPv4sec 对自己的数据包执行 PMTUD，并且如果 IPv4sec PMTU 发生更改（例如减小），则 IPv4sec 不会立即通知 GRE，但是，当另一个更大的数据包传入时，则会执行第 2 步中的过程。
4. GRE IPv4 MTU 现在更小，因此它会丢弃任何设置了 DF 位且现在对其而言过大的数据 IPv4 数据包，并向发送主机发送 ICMP 消息。

命令可以帮助 GRE 接口动态设置其 IPv4 MTU，而非如 命令一样实现静态设置。**tunnel path-mtu-discoveryip mtu**实际上，建议同时使用这两个命令。

命令用于为 GRE 和 IPv4sec 开销提供空间（相对于本地物理传出接口的 IPv4 MTU）。**ip mtu**

如果 IPv4sec 对等体之间的路径上存在 IPv4 MTU 更小的链路，则 命令允许 GRE 隧道 IPv4 MTU 进一步减小。**tunnel path-mtu-discovery**

如果在已配置 GRE + IPv4sec 隧道的网络中有 PMTUD 相关的问题，可以参考以下解决方案。

下面列表从最可取的解决方案开始列出。

1. 修复 PMTUD 无法正常工作的问题，这通常是由阻止 ICMP 的路由器或防火墙造成的。
2. 在隧道接口上使用 命令，使得路由器降低 TCP SYN 数据包的 TCP MSS 值。**ip tcp adjust-mss**这有助于两个终端主机（TCP 发送端和接收端）使用足够小的数据包，从而无需执行 PMTUD。
3. 在路由器的入口接口上使用策略路由并配置路由映射，从而在 DF 位传入 GRE 隧道接口之前，清除数据 IPv4 报头中的 DF 位。这使得数据 IPv4 数据包在 GRE 封装之前可以进行分段 。
4. 增加 GRE 隧道接口的“ip mtu”，使其等于出站接口 MTU。这使得数据 IPv4 数据包无需先分段就可以进行 GRE 封装。然后对 GRE 数据包执行 IPv4sec 加密和分段，再将其传出物理出站接口。在这种情况下，无需在 GRE 隧道接口上配置 命令。**tunnel path-mtu-discovery**由于 IPv4sec 对等体上的 IPv4 数据包重组在进程交换模式下进行，因此这可以极大减少吞吐量。

相关信息

- IP 路由支持页
- IPSec（IP 安全协议）支持页
- 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
- 技术支持和文档 - Cisco Systems

关于此翻译
思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。