Resolver fragmentación de IPv4 y problemas de MTU, MSS y PMTUD con GRE e IPSEC

Contenido

[Introducción](#)
[Antecedentes](#)
[Montaje y fragmentación de IPv4](#)
[Problemas con la fragmentación de IPv4](#)
[Evite la fragmentación de IPv4: Cómo funciona TCP MSS](#)
[Ejemplo 1](#)
[Ejemplo 2](#)
[¿Qué es PMTUD](#)
[Ejemplo 3](#)
[Ejemplo 4](#)
[Problemas con PMTUD](#)
[Topologías de Red Comunes que Necesitan PMTUD](#)
[Túnel](#)
[Consideraciones Sobre las Interfaces de Túnel](#)
[Router como participante de la PMTUD en el terminal del túnel](#)
[Ejemplo 5](#)
[Ejemplo 6](#)
[Modo de túnel IPsec puro](#)
[Ejemplo 7](#)
[Ejemplo 8](#)
[GRE e IPv4sec combinados](#)
[Ejemplo 9](#)
[Ejemplo 10](#)
[Más Recomendaciones](#)
[Información Relacionada](#)

Introducción

Este documento describe cómo funcionan la fragmentación de IPv4 y la detección de la unidad de transmisión máxima (PMTUD) de la ruta.

Antecedentes

También se analizan escenarios que implican el comportamiento de la PMTUD cuando se combina con diferentes combinaciones de túneles IPv4.

Montaje y fragmentación de IPv4

Aunque la longitud máxima de un datagrama IPv4 es de 65 535, la mayoría de los enlaces de transmisión imponen un límite inferior a la longitud máxima de paquetes, conocida como unidad máxima de transmisión (MTU). El valor de MTU depende del link de transmisión.

El diseño de IPv4 se adapta a las diferencias de MTU, ya que permite a los routers fragmentar los datagramas IPv4 según sea necesario.

La estación receptora es responsable del reensamblado de los fragmentos en el datagrama IPv4 original de tamaño completo.

La fragmentación de IPv4 divide un datagrama en partes que se vuelven a ensamblar más adelante.

Los campos de origen, destino, identificación, longitud total y desplazamiento de fragmentos de IPv4, junto con los indicadores "más fragmentos" (MF) y "no fragmentar" (DF) en el encabezado IPv4, se utilizan para la fragmentación y el reensamblado de IPv4.

Para obtener más información acerca de los mecanismos de montaje y fragmentación de IPv4, consulte RFC 791.

Esta imagen representa el diseño de un encabezado IPv4.

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

La identificación es de 16 bits y es un valor asignado por el remitente de un datagrama IPv4. Esto ayuda en el reensamblado de los fragmentos de un datagrama.

El desplazamiento del fragmento es de 13 bits e indica que pertenece al datagrama IPv4 original. Este valor es un múltiplo de 8 bytes.

Hay 3 bits para los indicadores de control en el campo flags del encabezado IPv4. El bit "no fragmentar" (DF) determina si se permite o no fragmentar un paquete.

El bit 0 está reservado y siempre se establece en 0.

El bit 1 es el bit DF (0 = "puede fragmentar", 1 = "no fragmentar").

El Bit 2 es el bit "more fragments" (MF) (0 = "último fragmento", 1 = "más fragmentos").

Valor	Bit 0 Reservado	Bit 1 DF	Bit 2 MF
0	0	Se puede	Último
1	0	No se puede	Más

Si se agregan las longitudes de los fragmentos de IPv4, el valor supera la longitud del datagrama IPv4 original en 60.

La razón por la que la longitud total aumenta por 60 es que se han creado tres encabezados IPv4 adicionales, uno para cada fragmento después del primer fragmento.

El primer fragmento tiene un desplazamiento de 0, la longitud de este fragmento es 1500; esto incluye 20 bytes para el encabezado IPv4 original ligeramente modificado.

El segundo fragmento tiene un desplazamiento de 185 (185 x 8 = 1480); la parte de datos de este fragmento comienza con 1480 bytes en el datagrama IPv4 original.

La longitud de este fragmento es 1500; esto incluye el encabezado IPv4 adicional creado para este fragmento.

El tercer fragmento tiene un desplazamiento de 370 (370 x 8 = 2960); la parte de datos de este fragmento comienza con 2960 bytes en el datagrama IPv4 original.

La longitud de este fragmento es 1500; esto incluye el encabezado IPv4 adicional creado para este fragmento.

El cuarto fragmento tiene un desplazamiento de 555 (555 x 8 = 4440), lo que significa que la parte de datos de este fragmento comienza con 4440 bytes en el datagrama IPv4 original.

La longitud de este fragmento es de 700 bytes; esto incluye el encabezado IPv4 adicional creado para este fragmento.

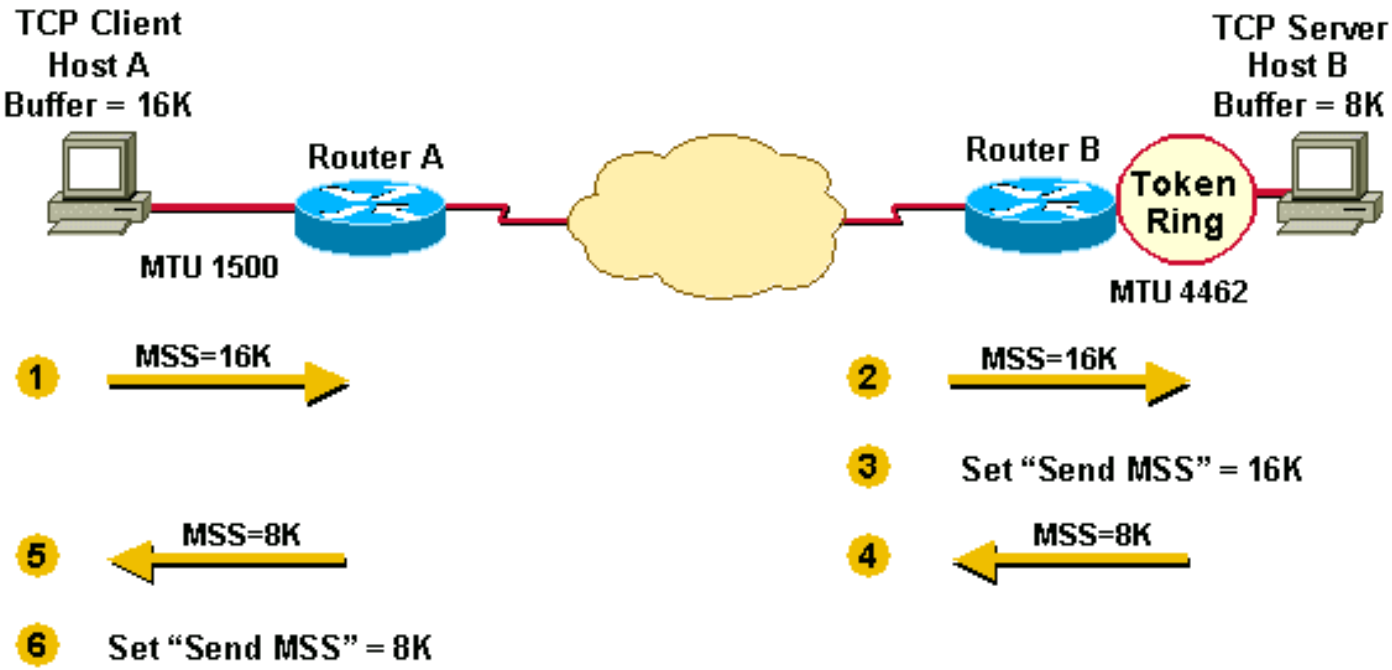
Es solo cuando se recibe el último fragmento que se puede determinar el tamaño del datagrama IPv4 original.

El desvío de fragmentos en el último fragmento (555) proporciona un desplazamiento de datos de 4440 bytes en el datagrama IPv4 original.

La suma de los bytes de datos del último fragmento (680 = 700 - 20) produce 5120 bytes, que es la parte de datos del datagrama IPv4 original.

La adición de 20 bytes para un encabezado IPv4 equivale al tamaño del datagrama IPv4 original

(4440 + 680 + 20 = 5140), como se muestra en las imágenes.



Problemas con la fragmentación de IPv4

La fragmentación de IPv4 produce un pequeño aumento de la sobrecarga de memoria y CPU para fragmentar un datagrama IPv4. Esto es cierto para el remitente y para un router en la trayectoria entre un remitente y un receptor.

La creación de fragmentos implica la creación de encabezados de fragmentos y copia el datagrama original en los fragmentos.

Esto se hace de manera eficiente porque la información necesaria para crear los fragmentos está disponible inmediatamente.

La fragmentación genera mayor sobrecarga para el receptor al reensamblar los fragmentos porque el receptor debe asignar memoria para los fragmentos que llegan y unirlos nuevamente en un datagrama una vez recibidos todos los fragmentos.

El reensamblado en un host no se considera un problema porque el host tiene el tiempo y los recursos de memoria para dedicarlos a esta tarea.

Sin embargo, el reensamblado es ineficiente en un router cuya tarea principal es reenviar los paquetes lo más rápido posible.

Un router no está diseñado para conservar los paquetes durante un período de tiempo.

Un router que realiza el reensamblado elige el búfer más grande disponible (18K), porque no tiene forma de determinar el tamaño del paquete IPv4 original hasta que se recibe el último fragmento.

Otro problema de la fragmentación radica en cómo se manejan los fragmentos descartados.

Si se descarta un fragmento de un datagrama IPv4, debe estar presente todo el datagrama IPv4

original y también está fragmentado.

Esto se observa con Network File System (NFS). NFS tiene un tamaño de bloque de lectura y escritura de 8192.

Por lo tanto, un datagrama IPv4/UDP de NFS es de aproximadamente 8500 bytes (que incluye encabezados NFS, UDP e IPv4).

Una estación de envío conectada a una Ethernet (MTU 1500) tiene que fragmentar el datagrama de 8500 bytes en seis (6) partes: cinco (5) fragmentos de 1500 bytes y un (1) fragmento de 1100 bytes.

Si alguno de los seis fragmentos se descarta debido a un link congestionado, el datagrama original completo debe ser retransmitido. Esto da como resultado la creación de seis fragmentos más.

Si este link descarta uno de cada seis paquetes, las probabilidades de que se transfieran datos NFS a través de este link son bajas, porque al menos un fragmento de IPv4 se descartaría de cada datagrama IPv4 original de 8500 bytes de NFS.

Los firewalls que filtran o manipulan paquetes basados en la información de la capa 4 (L4) a la capa 7 (L7) tienen problemas para procesar correctamente los fragmentos de IPv4.

Si los fragmentos de IPv4 están desordenados, un firewall bloquea los fragmentos no iniciales porque no llevan la información que coincide con el filtro de paquetes.

Esto significa que el host receptor no pudo volver a montar el datagrama IPv4 original.

Si el firewall está configurado para permitir fragmentos no iniciales con información insuficiente para que coincidan correctamente con el filtro, es posible que se produzca un ataque de fragmento no inicial a través del firewall.

Los dispositivos de red como Content Switch Engines dirigen los paquetes basados en la información de L4 a L7 y, si un paquete abarca varios fragmentos, el dispositivo tiene problemas para aplicar sus políticas.

Evite la fragmentación de IPv4: Cómo funciona TCP MSS

El tamaño máximo de segmento (MSS) del protocolo de control de transmisión (TCP) define la cantidad máxima de datos que acepta un host en un único datagrama TCP/IPv4.

Este datagrama TCP/IPv4 posiblemente esté fragmentado en la capa IPv4. El valor de MSS se envía como una opción de encabezado TCP solamente en los segmentos SYN de TCP.

Cada lado de una conexión TCP informa su valor de MSS al otro lado. El valor MSS no se negocia entre hosts.

Se requiere que el host remitente limite el tamaño de los datos en un solo segmento TCP a un valor inferior o igual al valor de MSS informado por el host receptor.

Originalmente, el MSS pretendía indicar cuán grande era un búfer (mayor o igual que 65 496 bytes) asignado a una estación de recepción para almacenar los datos del TCP contenidos en un único datagrama IPv4.

MSS era el segmento máximo de datos que el receptor TCP iba a aceptar. Este segmento TCP podría ser tan grande como 64K y fragmentado en la capa IPv4 para ser transmitido al host receptor.

El host de recepción reensamblará el datagrama IPv4 antes de pasar el segmento del TCP completado a la capa del TCP.

Cómo se establecen y utilizan los valores MSS para limitar los tamaños de segmento TCP y datagrama IPv4.

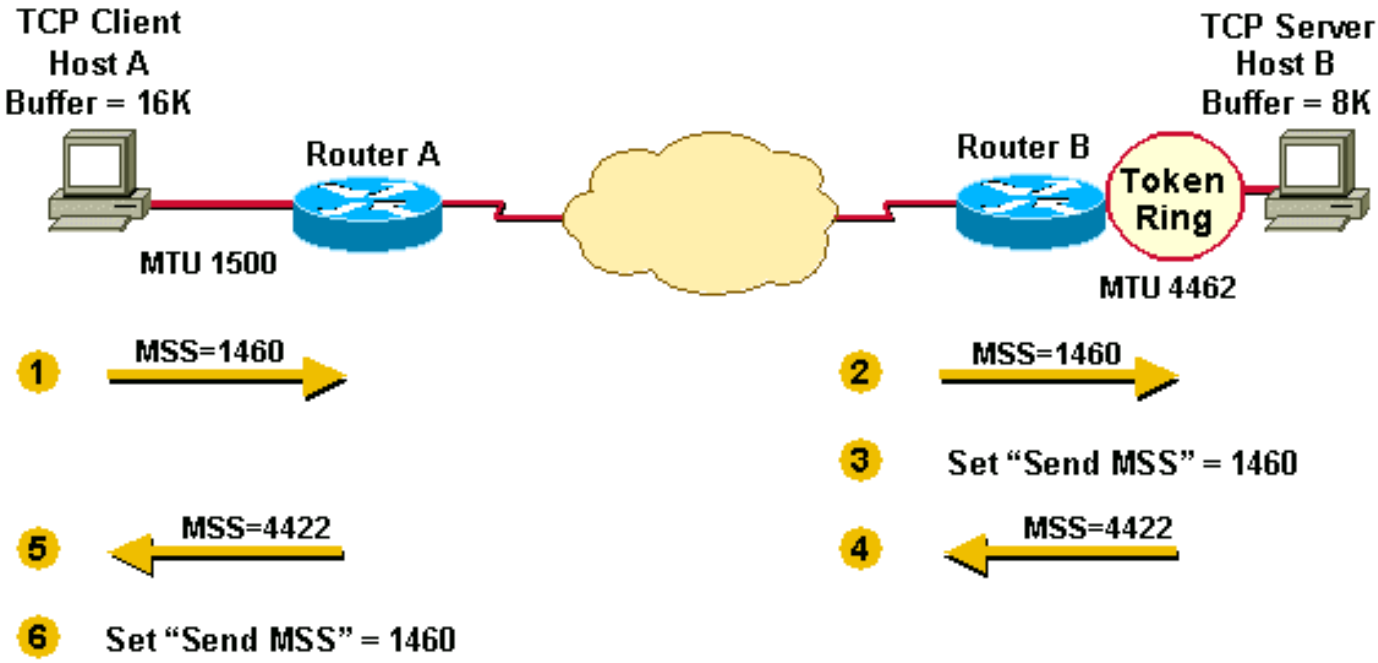
El ejemplo 1 ilustra la forma en que MSS se implementó por primera vez.

El Host A tiene un buffer de 16K y el Host B tiene un buffer de 8K. Envían y reciben sus valores MSS y ajustan sus envíos MSS para enviar información entre ellos.

El host A y el host B tienen que fragmentar los datagramas IPv4 que son más grandes que la MTU de la interfaz, pero menos que el MSS de envío porque la pila TCP pasa 16K u 8K bytes de datos por la pila a IPv4.

En el caso del Host B, los paquetes se fragmentan para llegar a la LAN Token Ring y nuevamente para llegar a la LAN Ethernet.

Ejemplo 1



1. El Host A envía su valor de MSS de 16K al Host B.
2. El Host B recibe el valor de MSS de 16K del Host A.
3. El Host B configura su valor de MSS de envío en 16K.

4. El Host B envía su valor de MSS de 8K al Host A.
5. El Host A recibe el valor de MSS de 8K del Host B.
6. El Host A configura su valor de MSS de envío en 8K.

Para ayudar a evitar la fragmentación de IPv4 en los extremos de la conexión TCP, se cambió la selección del valor MSS al tamaño mínimo de búfer y a la MTU de la interfaz saliente (- 40).

Los números MSS son 40 bytes más pequeños que los números MTU porque MSS (el tamaño de los datos TCP) no incluye el encabezado IPv4 de 20 bytes ni el encabezado TCP de 20 bytes.

MSS se basa en los tamaños de encabezado predeterminados; la pila de remitentes debe restar los valores adecuados para el encabezado IPv4 y el encabezado TCP depende de las opciones de TCP o IPv4 que se utilicen.

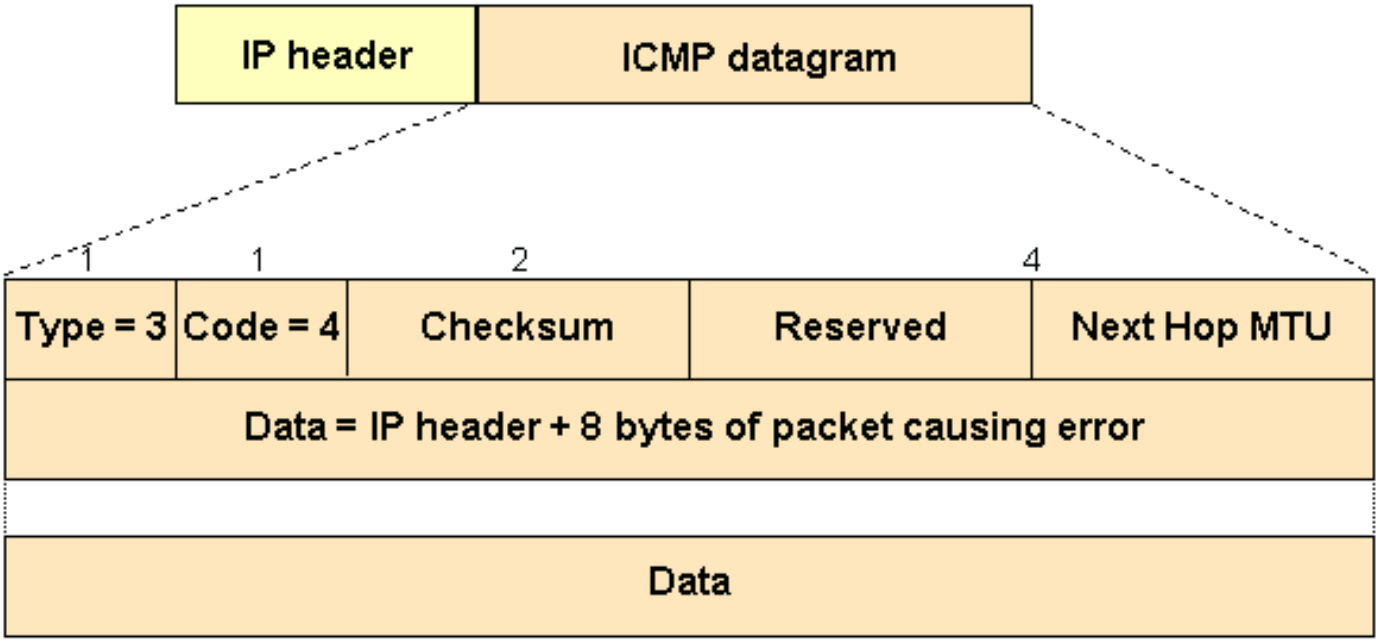
MSS funciona actualmente de una manera en la que cada host primero compara su MTU de interfaz saliente con su propio buffer y elige el valor más bajo como el MSS a enviar.

Los hosts luego comparan el tamaño de MSS recibido con su propia MTU de interfaz y nuevamente eligen el menor de los dos valores.

El ejemplo 2 ilustra este paso adicional realizado por el remitente para evitar la fragmentación en los cables locales y remotos.

Cada host tiene en cuenta la MTU de la interfaz saliente antes de que los hosts se envíen entre sí sus valores MSS. Esto ayuda a evitar la fragmentación.

Ejemplo 2



1. El Host A compara su buffer de MSS (16K) y su MTU (1500 - 40 = 1460) y utiliza el valor más bajo como el valor de MSS (1460) para enviarlo al Host B.
2. El Host B recibe el MSS de envío (1460) del Host A y lo compara con el valor de su MTU de

interfaz saliente - 40 (4422).
3. El host B establece el valor inferior (1460) como MSS para poder enviar datagramas IPv4 al host A.
4. El Host B compara su buffer de MSS (8K) y su MTU (4462 - 40 = 4422) y utiliza 4422 como el valor de MSS para enviarlo al Host A.
5. El Host A recibe el MSS de envío (4422) del Host B y lo compara con el valor de su interfaz saliente MTU -40 (1460).
6. El host A establece el valor inferior (1460) como MSS para poder enviar datagramas IPv4 al host B.

El valor elegido por ambos hosts como MSS de envío recíproco es 1460. A menudo, el valor de MSS de envío es el mismo en cada extremo de una conexión TCP.

En el ejemplo 2, la fragmentación no se produce en los extremos de una conexión TCP porque los hosts tienen en cuenta ambas MTU de interfaz saliente.

Los paquetes todavía se fragmentan en la red entre el router A y el router B si encuentran un link con una MTU inferior a la de la interfaz saliente de cualquiera de los hosts.

¿Qué es PMTUD

TCP MSS aborda la fragmentación en los dos extremos de una conexión TCP, pero no maneja los casos en los que hay un link de MTU más pequeño en el medio entre estos dos extremos.

PMTUD se desarrolló con el fin de evitar la fragmentación en la ruta entre los terminales. Se utiliza para determinar dinámicamente la MTU más baja a lo largo de la trayectoria desde un origen de paquete hasta su destino.

> Nota: PMTUD sólo es compatible con TCP y UDP. Otros protocolos no la admiten. Si la PMTUD está habilitada en un host, todos los paquetes TCP y UDP del host tienen el bit DF configurado.

Cuando un host envía un paquete de datos de MSS completo con el bit DF configurado, PMTUD reduce el valor de MSS de envío para la conexión si recibe información que indique que el paquete requiere fragmentación.

Un host registra el valor de MTU para un destino porque crea una entrada de host (/32) en su tabla de enrutamiento con este valor de MTU.

Si un router intenta reenviar un datagrama IPv4 (con el bit DF configurado) a un link que tiene una MTU inferior al tamaño del paquete, el router descarta el paquete y devuelve un mensaje "Destination Unreachable" (Destino inalcanzable) del Protocolo de mensajes de control de Internet (ICMP) al origen del datagrama IPv4 con el código que indica "fragmentación necesaria y DF configurado" (tipo 3, código 4).

Cuando la estación de origen recibe el mensaje ICMP, baja el MSS de envío, y cuando TCP retransmite el segmento, utiliza el tamaño de segmento más pequeño.

Este es un ejemplo de un mensaje ICMP "fragmentation needed and DF set" visto en un router después de la **debug ip icmp** comando está activado:

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

En este diagrama, se muestra el formato del encabezado ICMP de un mensaje de “destino inalcanzable” que dice “fragmentación necesaria y DF configurado”.

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

Según RFC 1191, un router que devuelve un mensaje de ICMP que indica "Fragmentación necesaria y DF configurado" debe incluir la MTU de esa red de siguiente salto en los 16 bits de orden inferior del campo de encabezado adicional del ICMP etiquetado como "no utilizado" en la especificación RFC 792.

Las primeras implementaciones de RFC 1191 no suministraban la información de MTU de salto siguiente. Incluso cuando esta información se suministraba, algunos hosts la ignoraban.

Para este caso, RFC 1191 también contiene una tabla que enumera los valores sugeridos por los cuales se reduce la MTU durante la PMTUD.

Es utilizado por los hosts para llegar más rápidamente a un valor razonable para el MSS de envío y como se muestra en este ejemplo.

La PMTUD se realiza continuamente en todos los paquetes porque la trayectoria entre el remitente y el receptor puede cambiar dinámicamente.

Cada vez que un remitente recibe un mensaje ICMP "No se puede fragmentar", actualiza la información de enrutamiento (donde almacena la PMTUD).

Durante PMTDU, pueden ocurrir dos cosas:

1. El paquete puede llegar hasta el receptor sin ser fragmentado.

> Nota: Para que un router proteja la CPU contra ataques DoS, limita el número de mensajes ICMP inalcanzables que enviaría a dos por segundo. Por lo tanto, en este contexto, si tiene un escenario de red en el que espera que el router deba responder con más de dos mensajes ICMP (tipo = 3, código = 4) por segundo (pueden ser hosts diferentes), inhabilite la aceleración de mensajes ICMP con el comando **no ip icmp rate-limit unreachable [df] interface** comando.

---

2. El remitente recibe mensajes ICMP "No se puede fragmentar" desde los saltos a lo largo de la trayectoria hacia el receptor.

PMTUD se realiza independientemente para ambas direcciones de un flujo de TCP.

Hay casos en los que la PMTUD en una dirección de un flujo activa una de las estaciones finales para reducir el MSS de envío y la otra estación final mantiene el MSS de envío original porque nunca envió un datagrama IPv4 lo suficientemente grande como para activar la PMTUD.

Un ejemplo es la conexión HTTP representada en el ejemplo 3. El cliente TCP envía paquetes pequeños y el servidor, paquetes grandes.

En este caso, sólo los paquetes grandes del servidor (mayores de 576 bytes) activan la PMTUD.

Los paquetes del cliente son pequeños (menos de 576 bytes) y no activan la PMTUD porque no requieren fragmentación para atravesar el link de MTU 576.

Ejemplo 3

El ejemplo 4 muestra un ejemplo de ruteo asimétrico donde una de las trayectorias tiene una MTU mínima menor que la otra.

El routing asimétrico se produce cuando se toman diferentes rutas para enviar y recibir datos entre dos terminales.

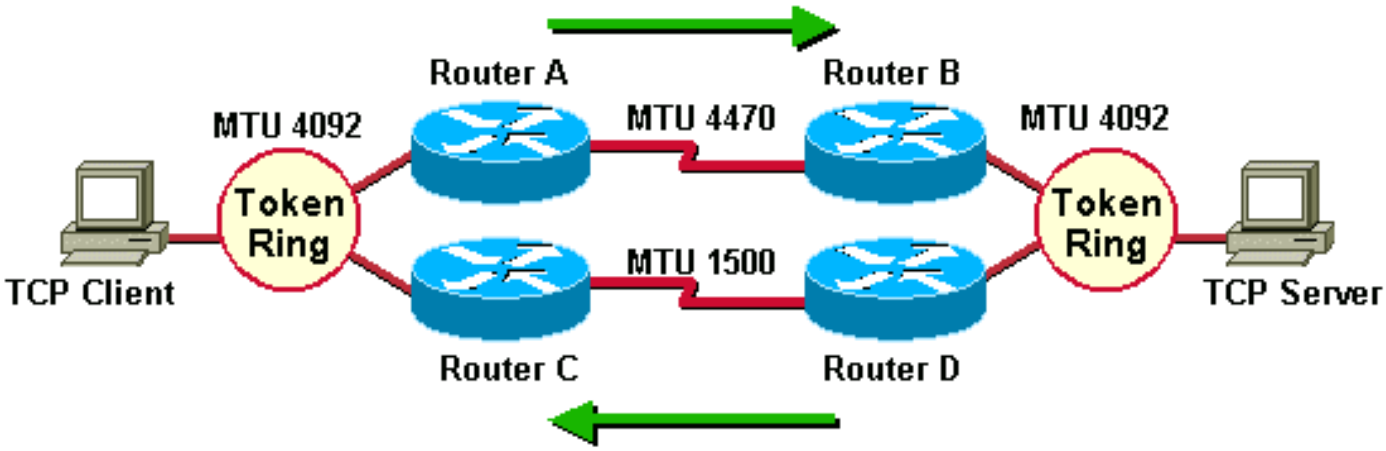
En este ejemplo, PMTUD activa la reducción del MSS de envío sólo en una dirección de un flujo TCP.

El tráfico del cliente TCP al servidor tiene lugar a través del router A y el router B, mientras que el tráfico de retorno proveniente del servidor y dirigido al cliente se transmite por el router D y el router C.

Cuando el servidor TCP envía paquetes al cliente, la PMTUD hace que el servidor reduzca el MSS de envío porque el Router D debe fragmentar los paquetes de 4092 bytes antes de poder enviarlos al Router C.

Por el contrario, el cliente nunca recibe un mensaje "Destination Unreachable" de ICMP con el código que indica "fragmentación necesaria y DF configurado" porque el router A no tiene que fragmentar paquetes cuando los envía al servidor a través del router B.

Ejemplo 4



Nota: El comando ip tcp path-mtu-discovery se utiliza para habilitar la detección de trayectoria de MTU TCP para las conexiones TCP iniciadas por routers (BGP y Telnet, por ejemplo).

Problemas con PMTUD

Estas son cosas que pueden interrumpir la PMTUD.

- Un router descarta un paquete y no envía un mensaje ICMP. (Poco común)
- Un router genera y envía un mensaje ICMP, pero el mensaje ICMP es bloqueado por un router o firewall entre este router y el remitente. (Común)

- Un router genera y envía un mensaje ICMP, pero el remitente ignora el mensaje. (Poco común)

La primera y la última de estas tres viñetas suelen ser el resultado de un error, pero la viñeta del medio describe un problema común.

Aquellos que implementan filtros de paquetes ICMP tienden a bloquear todos los tipos de mensajes ICMP en lugar de bloquear sólo ciertos tipos de mensajes ICMP.

Es posible que el filtro de paquetes bloquee todos los tipos de mensajes ICMP excepto aquellos que son "inalcanzables" o "excedidos en el tiempo".

El éxito o el fracaso de la PMTUD depende de los mensajes de ICMP inalcanzable que llegan al remitente de un paquete TCP/IPv4.

Los mensajes del ICMP que superan el tiempo son importantes para otros problemas de IPv4.

Aquí se muestra un ejemplo de ese filtro de paquetes implementado en un router.

```
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
```

Existen otras técnicas que se pueden utilizar para aliviar el problema de un ICMP completamente bloqueado.

- Borre el bit DF en el router y permita la fragmentación. (Aunque no es una buena idea. Consulte Problemas con la Fragmentación IP para obtener más información.
- Manipule el valor de la opción MSS de TCP con el comando interface **ip tcp adjust-mss <500-1460>.**

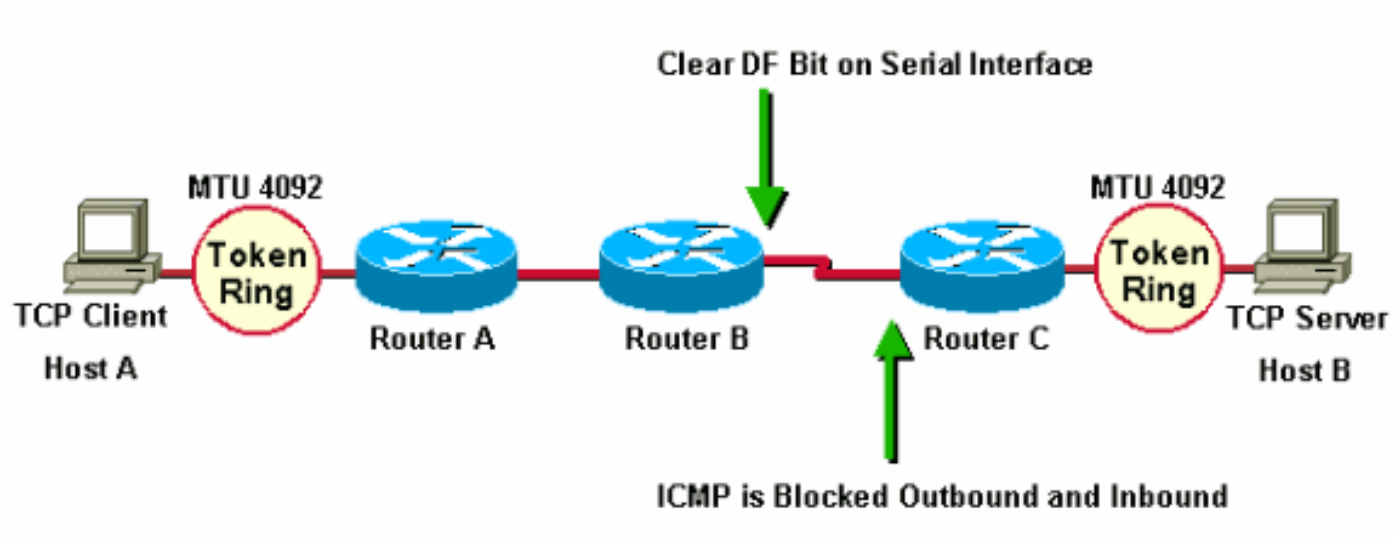
En el siguiente ejemplo, el Router A y el Router B están en el mismo dominio administrativo. No es posible acceder al router C y este bloquea el ICMP, por eso se interrumpe PMTUD.

Una solución para esta situación es borrar el bit DF en ambas direcciones en el router B para permitir la fragmentación. Es posible hacerlo con el routing de políticas.

La sintaxis para borrar el bit DF está disponible en el Cisco IOS® Software, versión 12.1(6) y posteriores.

```
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
```



Otra opción es cambiar el valor de opción de MSS del TCP en los paquetes de sincronización que cruzan el router (disponible en CISCO IOS® 12.2(4)T y posterior).

Esto reduce el valor de la opción MSS en el paquete TCP SYN para que sea menor que el valor (1460) en el **ip tcp adjust-mss**comando.

El resultado es que el remitente TCP envía segmentos no mayores que este valor.

El tamaño del paquete IPv4 es 40 bytes mayor (1500) que el valor MSS (1460 bytes) para dar cuenta del encabezado TCP (20 bytes) y el encabezado IPv4 (20 bytes).

Puede ajustar el MSS de los paquetes SYN TCP con el comando **ip tcp adjust-mss** comando. Esta sintaxis reduce el valor MSS en los segmentos TCP a 1460.

Este comando afecta el tráfico tanto entrante como saliente en la interfaz serial0.

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

Los problemas de fragmentación de IPv4 se han ampliado debido a que los túneles IPv4 se implementan más.

Los túneles causan más fragmentación porque la encapsulación del túnel agrega "sobrecarga" al tamaño de un paquete.

Por ejemplo, la adición de Generic Router Encapsulation (GRE) agrega 24 bytes a un paquete y, después de este aumento, el paquete debe fragmentarse porque es mayor que la MTU saliente.

Topologías de Red Comunes que Necesitan PMTUD

PMTUD se necesita en situaciones de red en las que los links intermedios tienen MTU más pequeñas que la MTU de los links extremos. Algunos motivos comunes para la existencia de estos links MTU más pequeños son:

- Token Ring (o FDDI): hosts extremos conectados con una conexión de Ethernet entre ellos. Las MTU de Token Ring (o interfaz de datos distribuidos por fibra, FDDI) en los extremos son superiores a la MTU de Ethernet en el medio.
- PPPoE (a menudo utilizado con ADSL) necesita un encabezado de 8 bytes. Esto reduce la MTU efectiva de Ethernet a 1492 (1500 - 8).

Los protocolos de túnel como GRE, IPv4sec y L2TP también necesitan espacio para sus respectivos encabezados y colas. Esto también reduce la MTU efectiva de la interfaz saliente.

Túnel

Un túnel es una interfaz lógica en un router de Cisco que proporciona una manera de encapsular los paquetes pasajeros dentro de un protocolo de transporte.

Es una arquitectura diseñada para proporcionar servicios a fin de implementar un esquema de encapsulamiento de punto a punto. Las interfaces de túnel tienen estos tres componentes principales:

- Protocolo de pasajero (AppleTalk, Banyan VINES, CLNS, DECnet, IPv4 o IPX)
- Protocolo de portadora. Uno de estos protocolos de encapsulamiento:
    - GRE: protocolo de portadora multiprotocolo de Cisco. Consulte RFC 2784 y RFC 1701 para obtener más información.
    - IPv4 en túneles IPv4; consulte RFC 2003 para obtener más información.
- Protocolo de transporte: protocolo utilizado para llevar a cabo el protocolo de encapsulamiento.

Los paquetes que se muestran en esta sección ilustran los conceptos de túneles IPv4 en los que el GRE es el protocolo de encapsulamiento e IPv4 es el protocolo de transporte.

El protocolo de pasajero también es IPv4. En este caso, IPv4 es tanto el protocolo de transporte como de pasajero.

Paquete Normal

IPv4	TCP	TELNET

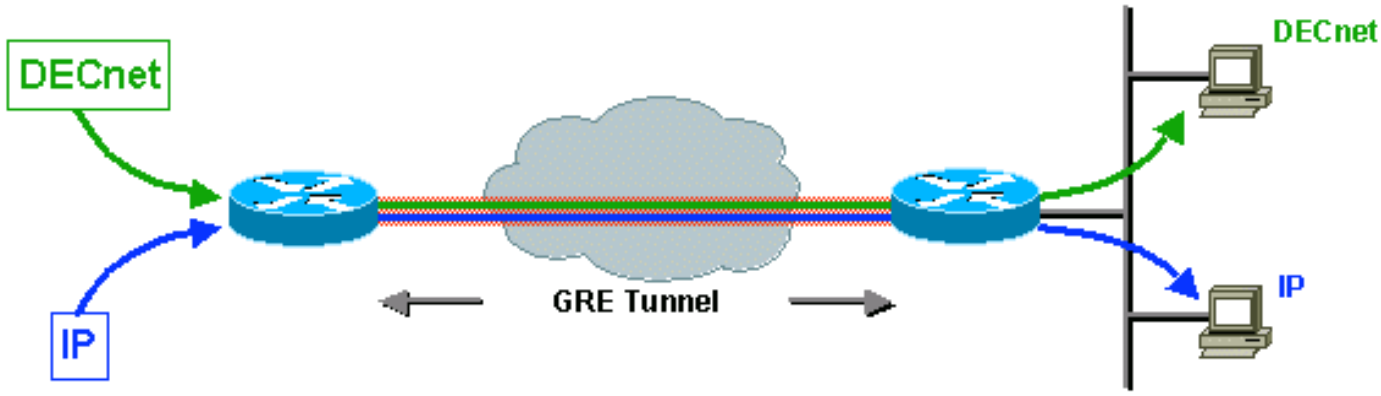
Paquete de Túnel

IPv4	GRE	IPv4	TCP	TELNET

- IPv4 es el protocolo de transporte.
- GRE es el protocolo de encapsulación.
- IPv4 es el protocolo de pasajero.

El ejemplo siguiente muestra el encapsulamiento de IPv4 y DECnet como protocolos de pasajero con GRE como portador.

Esto ilustra la posibilidad de que los protocolos de portadora encapsulen múltiples protocolos de pasajero como se muestra en la imagen.



Un administrador de red considera la tunelización en una situación en la que hay dos redes no IPv4 no contiguas separadas por una estructura básica IPv4.

Si las redes no contiguas ejecutan DECnet, el administrador puede optar por conectarlas (o no) configurando DECnet en la estructura básica.

El administrador no desea permitir que el ruteo DECnet consuma ancho de banda de la estructura básica porque esto podría interferir con el rendimiento de la red IPv4.

Una alternativa viable es hacer un túnel DECnet a través de la red troncal IPv4. La solución de túnel encapsula los paquetes DECnet dentro de IPv4 y los envía a través de la estructura básica al extremo del túnel donde se elimina la encapsulación y los paquetes DECnet se enrutan a su destino a través de DECnet.

Existen ventajas para encapsular el tráfico dentro de otro protocolo:

- Los terminales utilizan direcciones privadas (RFC 1918) y la red troncal no admite el enrutamiento de estas direcciones.
- Permita redes privadas virtuales (VPN) a través de WAN o Internet.
- Una las redes discontinuas de varios protocolos a través de una backbone de un solo protocolo.
- Cifre el tráfico a través de la backbone o Internet.

A partir de ahora, IPv4 se utilizará como protocolo pasajero e IPv4 como protocolo de transporte.

Consideraciones Sobre las Interfaces de Túnel

Estas son las consideraciones que deben tenerse en cuenta para los túneles.

- El cambio rápido de túneles GRE se introdujo en CISCO IOS® versión 11.1 y el cambio de CEF se introdujo en la versión 12.0.
- El CEF switching para los túneles GRE multipunto se introdujo en la versión 12.2(8)T.
- El encapsulamiento y desencapsulamiento en los terminales del túnel eran operaciones lentas en las versiones anteriores de Cisco IOS®, cuando solo se admitía el cambio de proceso.
- Hay problemas de topología y seguridad cuando se realiza la tunelización de paquetes. Los túneles pueden saltear listas de control de acceso (ACL) y firewalls.
- Si hace un túnel a través de un firewall, omite el protocolo pasajero que se está haciendo un túnel. Por lo tanto, se recomienda incluir la funcionalidad de firewall en los terminales del túnel con el fin de aplicar cualquier política en los protocolos de pasajero.
- La tunelización crea problemas con los protocolos de transporte que tienen temporizadores limitados (por ejemplo, DECnet) debido al aumento de la latencia.
- La tunelización a través de entornos con diferentes links de velocidad, como anillos FDDI rápidos y a través de líneas telefónicas lentas de 9600-bps, presenta problemas de reordenamiento de paquetes. Algunos protocolos pasajeros funcionan mal en redes de medios combinadas.
- Los túneles punto a punto consumen ancho de banda en un link físico. En varios túneles punto a punto, cada interfaz de túnel tiene un ancho de banda y la interfaz física sobre la que se ejecuta el túnel tiene un ancho de banda. Por ejemplo, establezca el ancho de banda del túnel en 100 Kb si había 100 túneles en un enlace de 10 Mb. El ancho de banda predeterminado para un túnel es 9 Kb.
- Los protocolos de ruteo prefieren un túnel sobre un link real porque el túnel aparenta ser un link de un salto con la trayectoria de menor costo, aunque involucra más saltos y por lo tanto más costoso que otra trayectoria. Esto se mitiga con una configuración adecuada del protocolo de ruteo. Considere la posibilidad de ejecutar un protocolo de ruteo diferente en la interfaz de túnel que el protocolo de ruteo que se ejecuta en la interfaz física.
- Los problemas con el ruteo recursivo se evitan al configurar las rutas estáticas apropiadas al destino del túnel. Una ruta recursiva es cuando la mejor ruta al destino del túnel se da a través del túnel mismo. Debido a esta situación, se devuelve la interfaz de túnel ("rebote") una y otra vez. Este error se observa cuando hay un problema de ruteo recursivo.

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

Router como participante de la PMTUD en el terminal del túnel

Cuando es el extremo de un túnel, el router tiene que realizar dos funciones de PMTUD diferentes.

- En la primera función, el router es el reenviador de un paquete del host. Para el procesamiento de PMTUD, el router debe verificar el bit DF y el tamaño de paquete del paquete de datos original, y realizar la acción apropiada, cuando sea necesario.
- La segunda función entra en juego después de que el router encapsula el paquete de IPv4 original dentro del paquete del túnel. En esta etapa, la función del router es más parecida a la de un host respecto de la PMTUD y el paquete de IPv4 del túnel.

Cuando el router actúa en la primera función (un router que reenvía paquetes IPv4 del host), esta función entra en juego antes de que el router encapsule el paquete IPv4 del host dentro del paquete del túnel.

Si el router participa como el reenviador de un paquete de host, completa estas acciones:

- Verificar si el bit DF está configurado.
- Verificar a qué tamaño de paquete puede adaptarse el túnel.
- Fragmento (si el paquete es demasiado grande y el bit DF no está configurado), encapsular fragmentos y enviar; o
- Descartar el paquete (si el paquete es demasiado grande y el bit DF está configurado) y enviar un mensaje de ICMP al remitente.
- Encapsular (si el paquete no es demasiado grande) y enviar.

De manera genérica, existe la opción de encapsular y fragmentar (enviar dos fragmentos de encapsulamiento) o de fragmentar y encapsular (enviar dos fragmentos encapsulados).

En esta sección se detallan dos ejemplos que muestran la interacción de la PMTUD y los paquetes que atraviesan las redes de ejemplo.

En el primer ejemplo, se observa lo que le sucede a un paquete cuando el router (en el origen del túnel) desempeña la función de router de reenvío.

Para procesar la PMTUD, el router necesita verificar el bit DF y el tamaño del paquete de datos original y tomar las medidas apropiadas.

Este ejemplo utiliza una encapsulación GRE para el túnel. GRE realiza la fragmentación antes de la encapsulación.

En los ejemplos posteriores, se muestran situaciones donde se realiza la fragmentación después de la encapsulación.

En el ejemplo 1, el bit DF no está configurado (DF = 0) y la MTU de IPv4 del túnel GRE es de 1476 (1500 - 24).

Ejemplo 1

1. El router de reenvío (en el origen del túnel) recibe un datagrama de 1500 bytes con el bit DF despejado (DF = 0) del host de envío.

Este datagrama está compuesto por un encabezado IP de 20 bytes más una carga útil TCP de 1480 bytes.

IPv4	TCP de 1480 bytes + datos

2. Dado que el paquete es demasiado grande para la MTU de IPv4 después de agregar la sobrecarga de GRE (24 bytes), el router de reenvío divide el datagrama en dos fragmentos de 1476 (encabezado IPv4 de 20 bytes + carga IPv4 de 1456 bytes) y 44 bytes (encabezado IPv4 de 20 bytes + carga IPv4 de 24 bytes)

Después de agregar la encapsulación GRE, el paquete no es mayor que la MTU de la interfaz física saliente.

IP0	TCP de 1456 bytes + datos
IP1	datos de 24 bytes

3. El router de reenvío agrega la encapsulación GRE, que incluye un encabezado GRE de 4 bytes más un encabezado IPv4 de 20 bytes, a cada fragmento del datagrama IPv4 original.

Estos dos datagramas IPv4 ahora tienen una longitud de 1500 y 68 bytes, y se interpretan como datagramas IPv4 individuales, no como fragmentos.

IPv4	GRE	IP0	TCP de 1456 bytes + datos
IPv4	GRE	IP1	datos de 24 bytes

4. El router de destino del túnel elimina la encapsulación GRE de cada fragmento del datagrama original, lo que deja dos fragmentos de IPv4 con longitudes de 1476 y 24 bytes.

Estos fragmentos de datagramas IPv4 se desvían por separado mediante el router al host de recepción.

IP0	TCP de 1456 bytes + datos
IP1	datos de 24 bytes

5. El host receptor reensambla estos dos fragmentos en el datagrama original.

IPv4	TCP de 1480 bytes + datos

El ejemplo 2 representa la función del router de reenvío en el contexto de una topología de red.

El router actúa en la misma función que el router de reenvío, pero esta vez se establece el bit DF

(DF = 1).

Ejemplo 2

1. El router de reenvío en el origen del túnel recibe un datagrama de 1500 bytes con DF = 1 del host de envío.

IPv4	TCP de 1480 bytes + datos

2. Dado que el bit DF está configurado y el tamaño del datagrama (1500 bytes) es mayor que la MTU IPv4 del túnel GRE (1476), el router descarta el datagrama y envía un mensaje "Fragmentación ICMP necesaria pero bit DF configurado" al origen del datagrama.

El mensaje ICMP alerta al remitente de que la MTU es 1476.

IPv4	MTU ICMP 1476

3. El host remitente recibe el mensaje ICMP y, cuando vuelve a enviar los datos originales, utiliza un datagrama IPv4 de 1476 bytes.

IPv4	TCP de 1456 bytes + datos

4. Esta longitud de datagrama IPv4 (1476 bytes) es ahora igual en valor a la MTU IPv4 del túnel GRE, por lo que el router agrega la encapsulación GRE al datagrama IPv4.

IPv4	GRE	IPv4	TCP de 1456 bytes + datos

5. El router receptor (en el destino del túnel) elimina la encapsulación GRE del datagrama IPv4 y la envía al host receptor.

IPv4	TCP de 1456 bytes + datos

Esto es lo que sucede cuando el router actúa en la segunda función como host de envío con respecto a la PMTUD y con respecto al paquete IPv4 del túnel.

Esta función entra en juego después de que el router haya encapsulado el paquete IPv4 original dentro del paquete de túnel.

> Nota: De forma predeterminada, un router no realiza la PMTUD en los paquetes de túnel GRE que genera. **tunnel path-mtu-discovery** se puede utilizar para activar la PMTUD para los paquetes de túnel GRE-IPv4.

En el ejemplo 3, se puede ver qué sucede cuando el host envía datagramas IPv4 que son lo suficientemente pequeños para caber en la MTU de IPv4 en la interfaz de túnel GRE.

El bit DF en este caso puede configurarse o borrarse (1 o 0).

La interfaz de túnel GRE no tiene el **tunnel path-mtu-discovery** configurado para que el router no cierre la PMTUD en el paquete GRE-IPv4.

Ejemplo 3

1. El router de reenvío en el origen del túnel recibe un datagrama de 1476 bytes del host de envío.

IPv4	TCP de 1456 bytes + datos

2. Este router encapsula el datagrama IPv4 de 1476 bytes dentro de GRE para obtener un datagrama IPv4 GRE de 1500 bytes.

Se borra el bit DF del encabezado IPv4 de GRE (DF = 0). Luego, este router reenvía este paquete al destino de túnel.

IPv4	GRE	IPv4	TCP de 1456 bytes + datos

3. Suponga que hay un router entre el origen y el destino del túnel con una MTU de link de 1400.

Este router fragmenta el paquete de túnel ya que el bit DF está despejado (DF = 0).

Recuerde que este ejemplo fragmenta el IPv4 más externo, por lo que los encabezados GRE, IPv4 más interno y TCP solo aparecen en el primer fragmento.

IP0	GRE	IP	TCP de 1352 bytes + datos
IP1	Datos de 104 bytes		

4. El router de destino del túnel debe reensamblar el paquete de túnel GRE.

IP	GRE	IP	TCP de 1456 bytes + datos

5. Después de que el paquete de túnel GRE se reensambla, el router quita el encabezado IPv4 GRE y envía el datagrama IPv4 original en su camino.

IPv4	TCP de 1456 bytes + datos

El ejemplo 4 muestra lo que sucede cuando el router actúa en la función de un host de envío con respecto a la PMTUD y con respecto al paquete IPv4 del túnel.

Esta vez el bit DF está configurado (DF = 1) en el encabezado IPv4 original y el **tunnel path-mtu-discovery** se ha configurado para que el bit DF se copie del encabezado IPv4 interno al encabezado externo (GRE + IPv4).

Ejemplo 4

1. El router de reenvío en el origen del túnel recibe un datagrama de 1476 bytes con DF = 1 del host de envío.

IPv4	TCP de 1456 bytes + datos

2. Este router encapsula el datagrama IPv4 de 1476 bytes dentro de GRE para obtener un

datagrama IPv4 GRE de 1500 bytes.

Este encabezado IPv4 de GRE tiene el conjunto de bits DF (DF = 1) ya que el datagrama IPv4 original tenía el conjunto de bits DF configurado.

Luego, este router reenvía este paquete al destino de túnel.

IPv4	GRE	IPv4	TCP de 1456 bytes

3. De nuevo, suponga que hay un router entre el origen y el destino del túnel con una MTU de link de 1400.

Este router no fragmenta el paquete de túnel porque el bit DF está configurado (DF=1).

Este router debe descartar el paquete y enviar un mensaje de error ICMP al router de origen del túnel, porque esa es la dirección IPv4 de origen en el paquete.

IPv4	MTU ICMP 1400

4. El router de reenvío en el origen del túnel recibe este mensaje de error "ICMP" y reduce la MTU IPv4 del túnel GRE a 1376 (1400 - 24).

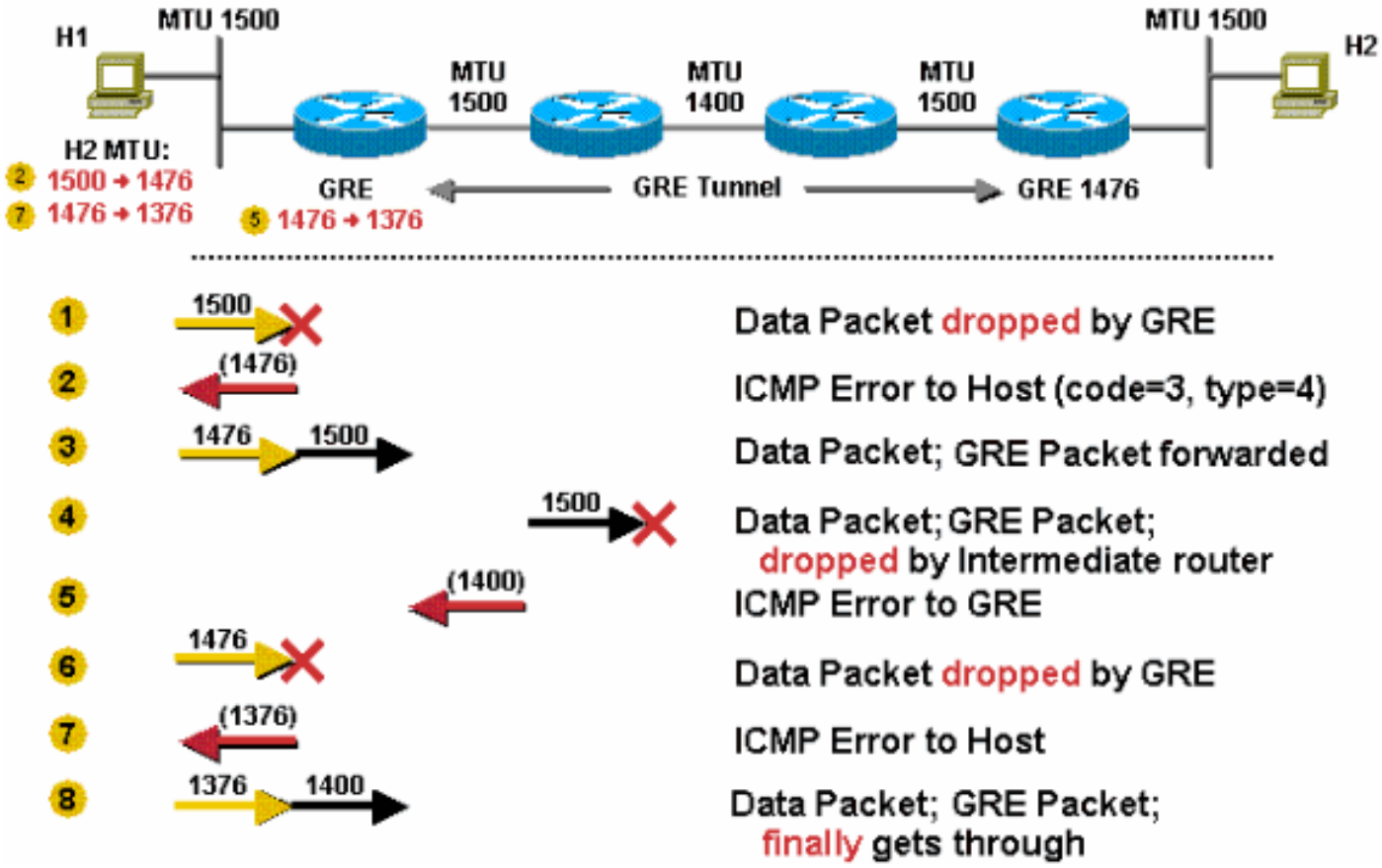
La próxima vez que el host remitente retransmita los datos en un paquete IPv4 de 1476 bytes, este paquete puede ser demasiado grande y este router envía un mensaje de error "ICMP" al remitente con un valor de MTU de 1376.

Cuando el host de envío retransmite los datos, los envía en un paquete IPv4 de 1376 bytes y este paquete atraviesa el túnel GRE hasta el host de recepción.

Ejemplo 5

Este ejemplo ilustra la fragmentación de GRE. Fragmente antes de la encapsulación para GRE, luego realice la PMTUD para el paquete de datos y el bit DF no se copia cuando el paquete IPv4 es encapsulado por GRE.

El bit DF no está configurado. La MTU de IPv4 de la interfaz del túnel GRE es, de forma predeterminada, 24 bytes menor que la MTU de IPv4 de la interfaz física, por lo que la MTU de IPv4 de la interfaz de GRE es de 1476, como se muestra en la imagen.



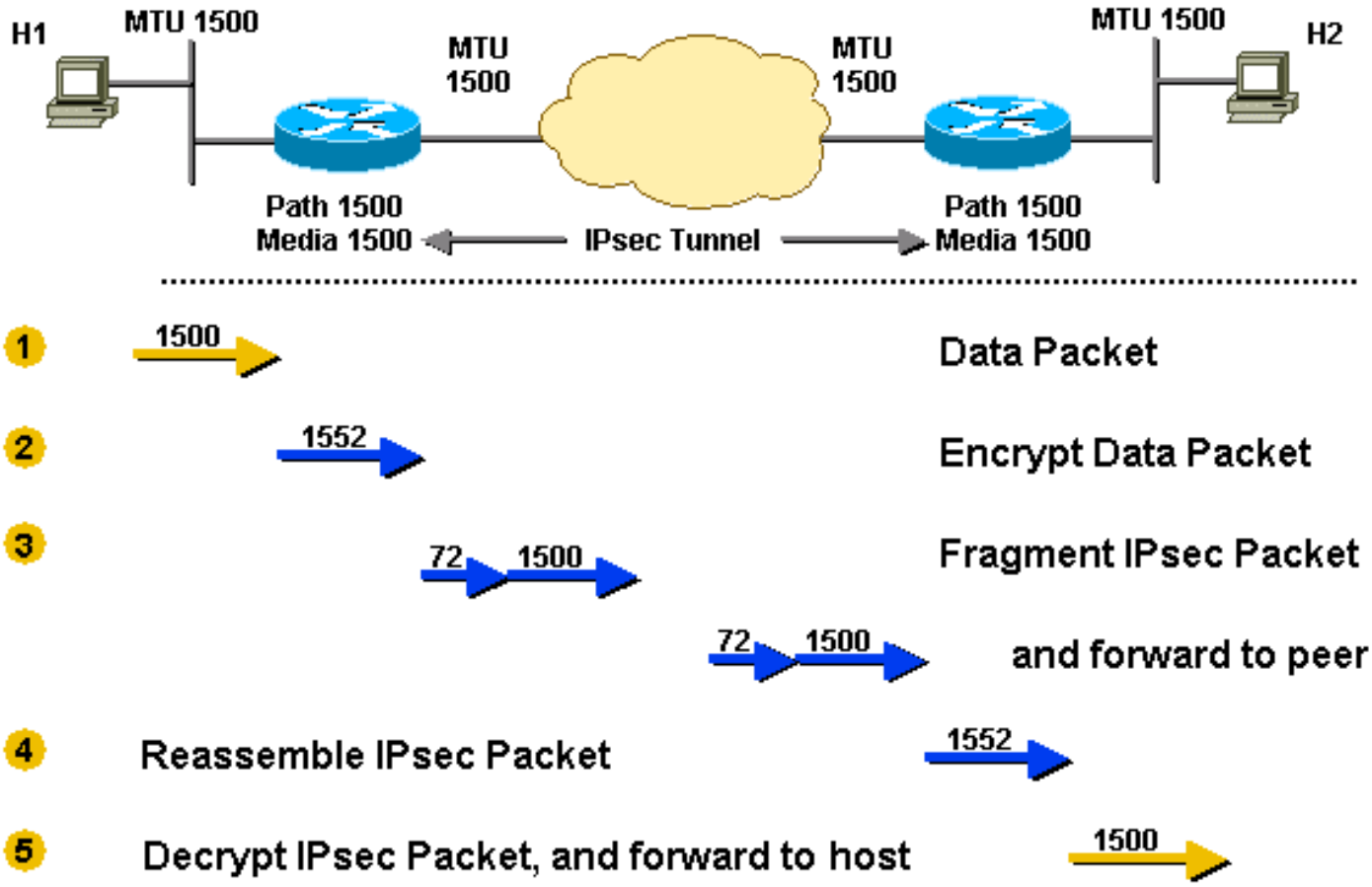
1. El emisor envía un paquete de 1500 bytes (encabezado IPv4 de 20 bytes + carga útil del TCP de 1480 bytes).
2. Debido a que la MTU del túnel GRE es 1476, el paquete de 1500 bytes se divide en dos fragmentos de IPv4 de 1476 y 44 bytes, cada uno anticipándose a los 24 bytes adicionales del encabezado GRE.
3. Los 24 bytes del encabezado GRE se agregan a cada fragmento de IPv4. Ahora, los fragmentos son de 1500 (1476 + 24) y 68 (44 + 24) bytes cada uno.
4. Los paquetes de GRE + IPv4 que contienen los dos fragmentos de IPv4 se reenvían al router de pares del túnel GRE.
5. El router de peer de túnel GRE quita los encabezados GRE de los dos paquetes.
6. Este router reenvía ambos paquetes al host de destino.
7. El host de destino vuelve a montar los fragmentos de IPv4 en el datagrama IPv4 original.

Ejemplo 6

Este ejemplo es similar al Ejemplo 5, pero esta vez se establece el bit DF. El router está configurado para realizar la PMTUD en paquetes de túnel GRE + IPv4 con el **tunnel path-mtu-discovery**y el bit DF se copia del encabezado IPv4 original al encabezado IPv4 de GRE.

Si el router recibe un error de ICMP para el paquete de GRE + IPv4, reduce la MTU de IPv4 en la interfaz del túnel GRE.

La MTU IPv4 del túnel GRE está configurada en 24 bytes menos que la MTU de la interfaz física de forma predeterminada, por lo que la MTU IPv4 de GRE aquí es 1476. Hay un link de MTU 1400 en la trayectoria del túnel GRE, como se muestra en la imagen.



1. El router recibe un paquete de 1500 bytes (encabezado IPv4 de 20 bytes + carga útil del TCP de 1480 bytes) y descarta el paquete. El router descarta el paquete porque es más grande que la MTU de IPv4 (1476) en la interfaz de túnel GRE.
2. El router envía un error de ICMP al remitente comunicándole que la MTU de salto siguiente es 1476. El host registra esta información, generalmente como una ruta de host para el destino en su tabla de ruteo.
3. El host remitente utiliza un tamaño de paquete de 1476 bytes cuando reenvía los datos. El router GRE agrega 24 bytes de encapsulación GRE y envía un paquete de 1500 bytes.
4. El paquete de 1500 bytes no puede atravesar el link de 1400 bytes y, por eso, será descartado por el router intermedio.
5. El router intermedio envía un ICMP (tipo = 3, código = 4) al router GRE con una MTU de siguiente salto de 1400. El router GRE reduce esto a 1376 (1400 - 24) y establece un valor de la MTU de IPv4 interna en la interfaz de GRE. Este cambio solo se puede ver cuando se utiliza el **debug tunnel** comando; no se puede ver en el resultado de la **show ip interface tunnel**<#> comando.
6. La próxima vez que el host reenvíe el paquete de 1476 bytes, el router GRE descarta el paquete, ya que es mayor que la MTU IPv4 actual (1376) en la interfaz de túnel GRE.
7. El router GRE envía otro ICMP (tipo = 3, código = 4) al remitente con una MTU de siguiente salto de 1376 y el host actualiza su información actual con un nuevo valor.
8. El host vuelve a enviar los datos, pero ahora en un paquete más pequeño de 1376 bytes, GRE agrega 24 bytes de encapsulación y los reenvía. Esta vez el paquete llega al par de túnel GRE, donde el paquete se desencapsula y se envía al host de destino.

Nota: Si el **tunnel path-mtu-discovery** El comando no se configuró en el router de reenvío en esta situación y el bit DF se configuró en los paquetes reenviados a través del túnel GRE. El host 1 sigue enviando correctamente los paquetes TCP/IPv4 al host 2, pero se fragmentan en el medio en el enlace de MTU 1400. Además, el par de túnel GRE debe reensamblarlos antes de que pueda desencapsularlos y reenviarlos.

Modo de túnel IPsec puro

El protocolo de seguridad IPv4 (IPv4sec) es un método basado en normas que otorga privacidad, integridad y autenticidad a la información transferida por las redes IPv4.

IPv4sec proporciona cifrado de capa de red IPv4. IPv4sec alarga el paquete de IPv4 agregando al menos un encabezado IPv4 (modo de túnel).

La longitud de los encabezados agregados varía en función del modo de configuración de IPv4sec, pero no superan los ~58 bytes (carga de seguridad de encapsulación (ESP) y autenticación ESP (ESPauth)) por paquete.

IPv4sec tiene dos modos: el modo de túnel y el modo de transporte.

1. El modo de túnel es el modo predeterminado. Con el modo de túnel, todo el paquete de IPv4 original está protegido (cifrado, autenticado o ambos) y se encapsula mediante los encabezados y colas de IPv4sec. Luego, se antepone un nuevo encabezado IPv4 al paquete que especifica los terminales de IPv4sec (pares) como origen y destino. El modo de túnel puede utilizarse con cualquier tráfico de IPv4 de unidifusión y debe usarse si IPv4sec protege el tráfico de los hosts detrás de los pares IPv4sec. Por ejemplo, se utiliza el modo de túnel con las redes privadas virtuales (VPN) donde los hosts de una red protegida envían paquetes a los hosts de una red protegida diferente a través de pares IPv4sec. Con las VPN, el "túnel" IPv4sec protege el tráfico de IPv4 entre los hosts mediante el cifrado del tráfico entre los routers de pares IPv4sec.
2. Con el modo de transporte (configurado con el subcomando, **mode transport**, en la definición de transformación), sólo se protege la carga del paquete IPv4 original (cifrada, autenticada o ambas). La carga útil se encapsula mediante los encabezados y colas de IPv4sec. Los encabezados IPv4 originales permanecen intactos, excepto en el campo de protocolo IPv4 que cambia a ESP (50), y el valor del protocolo original se guarda en la cola de IPv4sec para restaurarse cuando se descifre el paquete. El modo de transporte solo se utiliza cuando el tráfico de IPv4 que se protegerá se encuentra entre los pares IPv4sec; las direcciones IPv4 de origen y destino en el paquete son las mismas que las direcciones del par IPv4sec. Normalmente, el modo de transporte IPv4sec solo se utiliza cuando se usa otro protocolo de túnel (como GRE) para encapsular en primer lugar el paquete de datos de IPv4 y, a continuación, IPv4sec se utiliza para proteger los paquetes del túnel GRE.

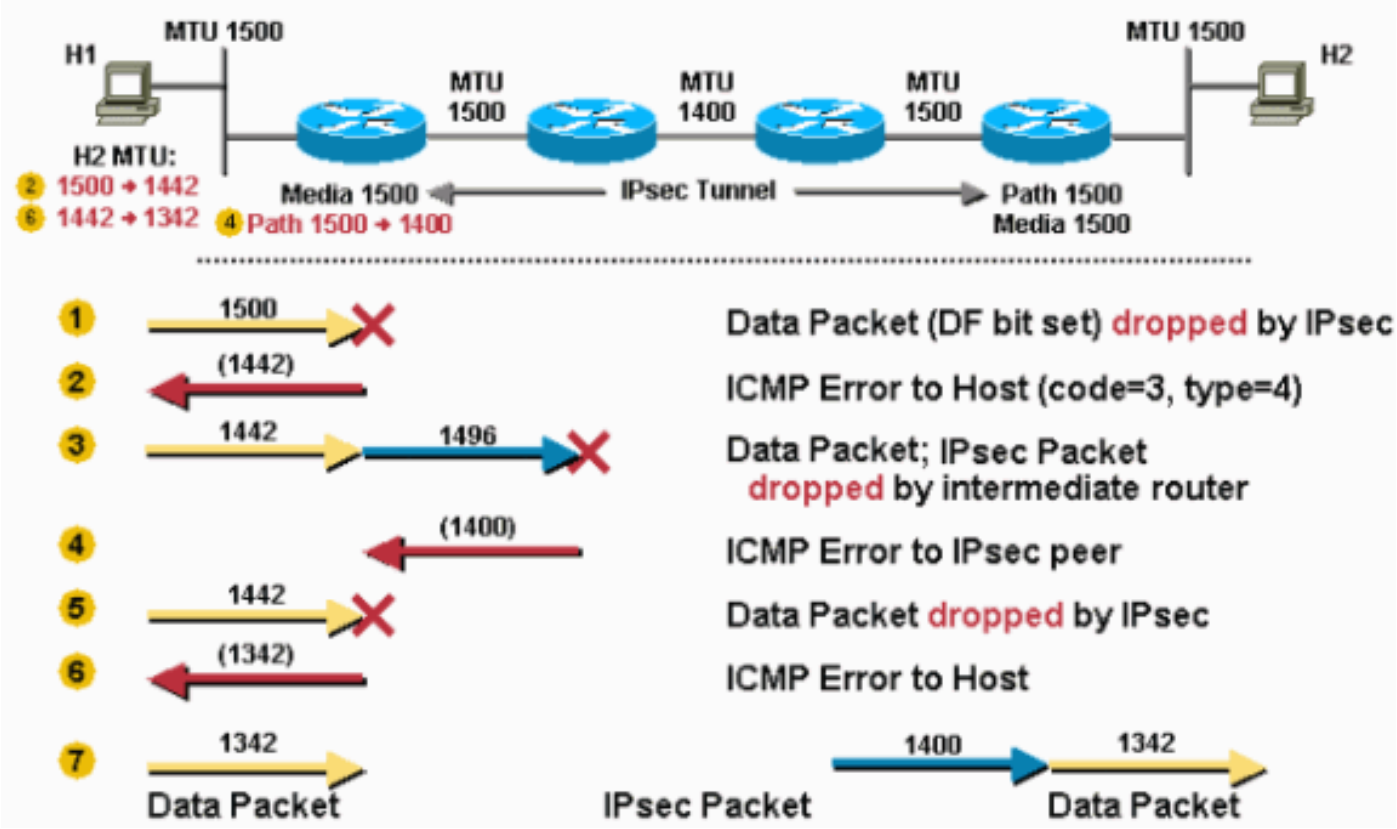
IPv4sec siempre realiza la PMTUD de los paquetes de datos y de sus propios paquetes. Hay comandos de configuración IPv4sec para modificar el procesamiento de la PMTUD para el paquete de IPv4 de IPv4sec; IPv4sec puede borrar, establecer o copiar el bit DF del encabezado IPv4 del paquete de datos en el encabezado IPv4 de IPv4sec. Esta función se denomina

"funcionalidad de invalidación del bit DF".

> Nota: Evite la fragmentación después de la encapsulación cuando el cifrado de hardware con IPv4sec se realiza. El cifrado de hardware ofrece un rendimiento de aproximadamente 50 Mbs, que depende del hardware, pero si el paquete IPv4sec se fragmenta, se pierde entre el 50 y el 90% del rendimiento. Esta pérdida se debe a que se cambia el proceso de los paquetes de IPv4sec fragmentados para el montaje y, a continuación, pasan al motor de cifrado de hardware para el descifrado. Esta pérdida de rendimiento puede bajar el rendimiento del cifrado del hardware al nivel de rendimiento del cifrado del software (2 - 10 MB).

Ejemplo 7

Esta situación representa la fragmentación de IPv4sec en acción. En esta situación, la MTU junto con la trayectoria entera es 1500. En esta situación, el bit DF no está configurado.



1. El router recibe un paquete de 1500 bytes (encabezado IPv4 de 20 bytes + carga útil del TCP de 1480 bytes) destinado al host 2.
2. El paquete de 1500 bytes se cifra mediante IPv4sec y se agregan 52 bytes de sobrecarga (encabezado IPv4sec, cola y encabezado IPv4 adicional). Ahora IPv4sec debe enviar un paquete de 1552 bytes. Dado que la MTU saliente es 1500, este paquete debe fragmentarse.
3. Se crean dos fragmentos fuera del paquete de IPv4sec. Durante la fragmentación, se agrega un encabezado IPv4 de 20 bytes adicional para el segundo fragmento, que da como resultado un fragmento de 1500 bytes y un fragmento IPv4 de 72 bytes.

4. El router de par del túnel IPv4sec recibe los fragmentos, elimina el encabezado IPv4 adicional y fusiona los fragmentos de IPv4 nuevamente en el paquete de IPv4sec original. A continuación, IPv4sec descifra este paquete.
5. Luego, el router reenvía el paquete de datos original de 1500 bytes al Host 2.

Ejemplo 8

Este ejemplo es similar al ejemplo 6, excepto en que en este caso el bit DF está configurado en el paquete de datos original y hay un link en la trayectoria entre los peers de túnel IPv4sec que tiene una MTU más baja que los otros links.

En este ejemplo se muestra cómo el router de peer IPv4sec realiza ambas funciones de PMTUD, como se describe en la sección El router como participante de PMTUD en el extremo de un túnel.

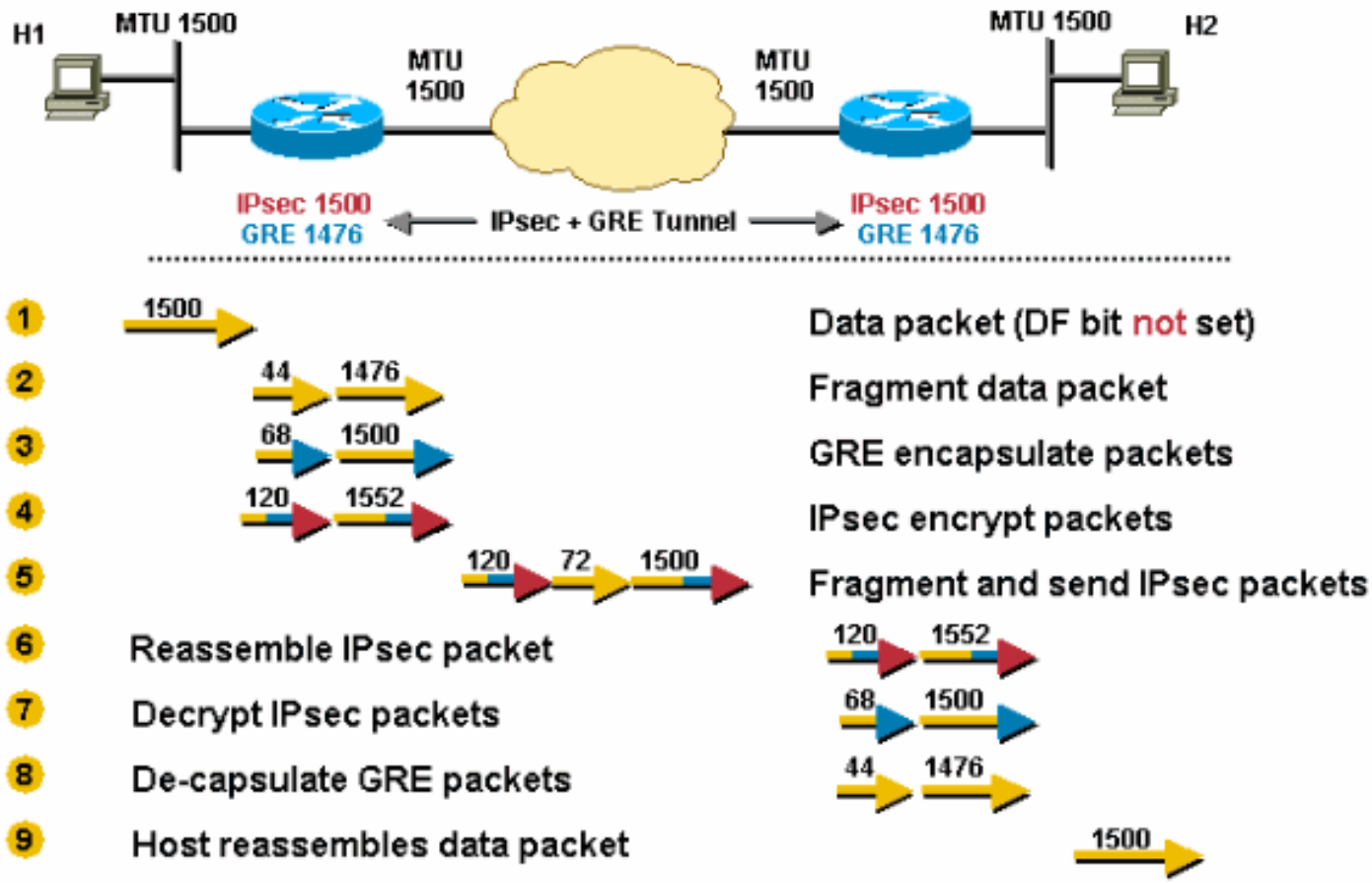
La PMTU de IPv4sec cambia a un valor inferior como resultado de la necesidad de fragmentación.

El bit DF se copia del encabezado IPv4 interno al encabezado IPv4 externo cuando IPv4sec cifra un paquete.

Los valores de la PMTU y la MTU de los medios se almacenan en la asociación de seguridad (SA) de IPv4sec.

La MTU de los medios se basa en la MTU de la interfaz de router saliente y la PMTU se basa en la MTU mínima detectada en la ruta entre los pares IPv4sec.

IPv4sec encapsula/cifra el paquete antes de intentar fragmentarlo, como se muestra en la imagen.



1. El router recibe un paquete de 1500 bytes y lo descarta porque la sobrecarga de IPv4sec, cuando se agrega, hace que el paquete sea más grande que la PMTU (1500).
2. El router envía un mensaje de ICMP al Host 1 comunicándole que la MTU de salto siguiente es 1442 (1500 - 58 = 1442). Estos 58 bytes son la sobrecarga máxima de IPv4sec cuando se utiliza IPv4sec ESP y ESPauth. La sobrecarga real de IPv4sec es posiblemente 7 bytes menor que este valor. El Host 1 registra esta información, generalmente como una ruta de host para el destino (Host 2), en su tabla de ruteo.
3. El Host 1 reduce su PMTU para el Host 2 a 1442, de modo que el Host 1 envía paquetes más pequeños (1442 bytes) cuando retransmite los datos al Host 2. El router recibe el paquete de 1442 bytes e IPv4sec agrega 52 bytes de sobrecarga de cifrado para que el paquete de IPv4sec resultante sea de 1496 bytes. Porque este paquete tiene el bit DF configurado en su encabezado, es descartado por el router del medio con el link de MTU de 1400 bytes.
4. El router central que descarta el paquete envía un mensaje de ICMP al remitente del paquete de IPv4sec (el primer router), indicando que la MTU de siguiente salto es de 1400 bytes. Este valor se registra en la PMTU de SA de IPv4sec.
5. La próxima vez que el Host 1 retransmita el paquete de 1442 bytes (no recibió un reconocimiento para él), IPv4sec descartará el paquete. El router descarta el paquete porque la sobrecarga de IPv4sec, cuando se agrega al paquete, lo hace más grande que la PMTU (1400).
6. El router envía un mensaje ICMP al Host 1 diciéndole que la MTU del salto siguiente es ahora 1342. (1400 - 58 = 1342). El host 1 vuelve a registrar esta información.
7. Cuando el Host 1 vuelve a retransmitir los datos, utiliza el paquete de menor tamaño (1342). Este paquete no requiere fragmentación y pasa a través del túnel IPv4sec al Host 2.

GRE e IPv4sec combinados

Cuando se usa IPv4sec para cifrar túneles GRE, tienen lugar interacciones más complejas para la fragmentación y PMTUD.

IPv4sec y GRE se combinan de esta manera porque IPv4sec no admite paquetes de multidifusión de IPv4, lo que significa que no es posible ejecutar un protocolo de routing dinámico por la VPN IPv4sec.

Los túneles GRE admiten la multidifusión a fin de utilizarse para encapsular en primer lugar el paquete de multidifusión del protocolo de routing dinámico en un paquete de unidifusión de IPv4 de GRE que puede cifrarse mediante IPv4sec.

Al hacer esto, IPv4sec se despliega a menudo en modo de transporte sobre GRE porque los peers de IPv4sec y los terminales del túnel GRE (los routers) son los mismos, y el modo de transporte ahorra 20 bytes de sobrecarga de IPv4sec.

Un caso interesante es cuando un paquete de IPv4 se divide en dos fragmentos y se encapsula mediante el GRE.

En este caso, IPv4sec detecta dos paquetes GRE + IPv4 independientes. A menudo, en una configuración predeterminada, uno de estos paquetes es lo suficientemente grande como para que deba fragmentarse después de haber sido cifrado.

El par IPv4sec debe volver a ensamblar este paquete antes del descifrado. Esta "doble fragmentación" (una vez antes del GRE y nuevamente tras IPv4sec) en el router de envío aumenta la latencia y disminuye el rendimiento.

El reensamblado es conmutado por proceso, por lo que hay un impacto de CPU en el router receptor siempre que esto sucede.

Esta situación puede evitarse configurando una "ip mtu" en la interfaz de túnel GRE lo suficientemente baja para tener en cuenta la sobrecarga del GRE y IPv4sec (de forma predeterminada, "ip mtu" en la interfaz de túnel GRE se establece en los bytes de sobrecarga de MTU/GRE de la interfaz real de salida).

Esta tabla enumera los valores de MTU sugeridos para cada combinación de túnel/modo que asume que la interfaz física saliente tiene una MTU de 1500.

Combinación de Túneles	MTU Específica Necesaria	MTU Recomendada
GRE + IPv4sec (modo de transporte)	1440 bytes	1400 bytes
GRE + IPv4sec (modo de túnel)	1420 bytes	1400 bytes

> Nota: Se recomienda el valor de MTU de 1400 porque cubre las combinaciones de modo GRE + IPv4sec más comunes. Además, no hay desventaja notable en permitir una sobrecarga adicional de 20 o 40 bytes. Es más fácil recordar y configurar un valor y este valor cubre casi todas las situaciones.

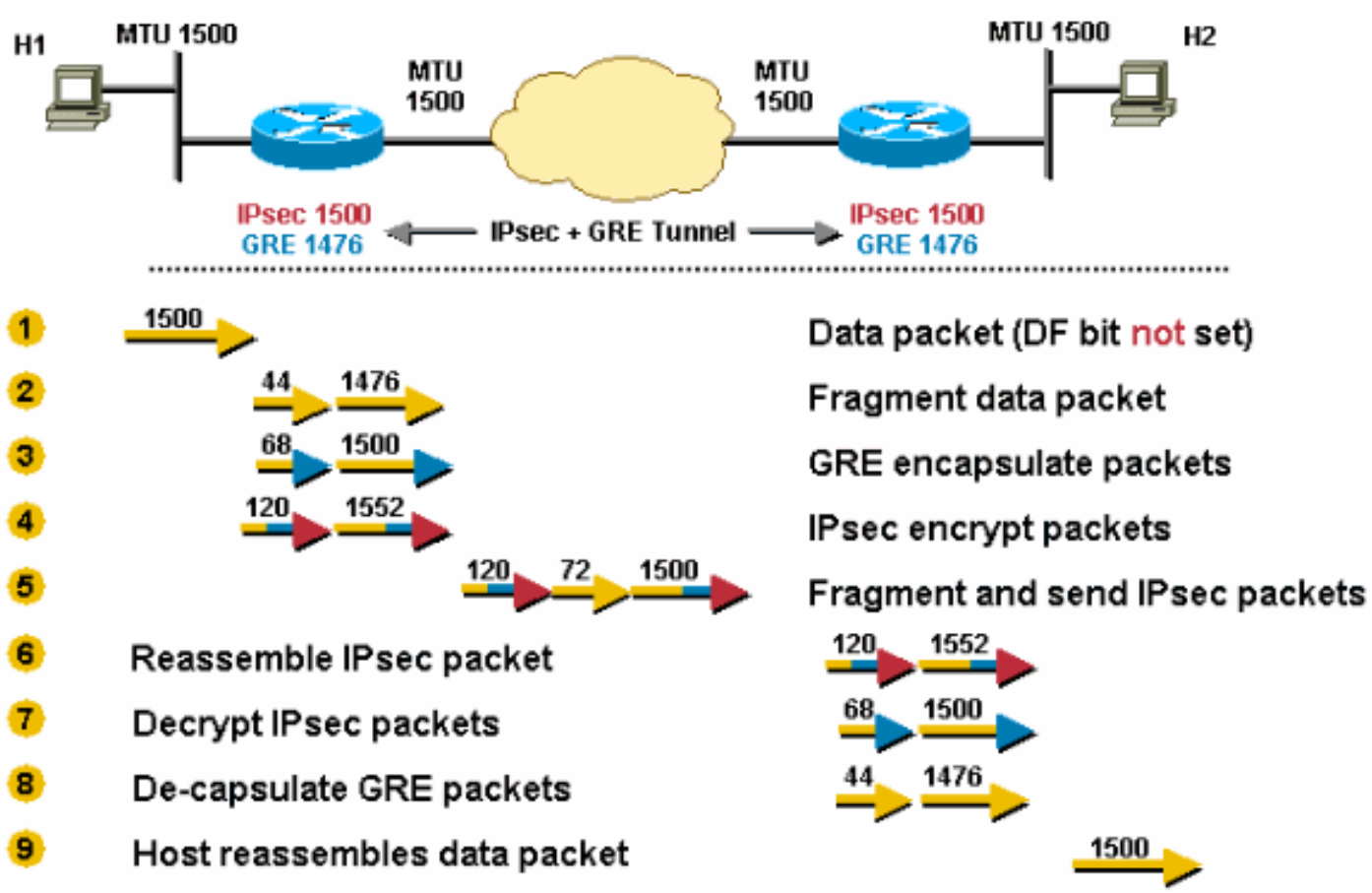
Ejemplo 9

IPv4sec se implementa en la parte superior del GRE. La MTU física saliente es de 1500, la PMTU de IPv4sec es de 1500 y la MTU de IPv4 del GRE es de 1476 (1500 - 24 = 1476).

Por lo tanto, los paquetes TCP/IPv4 se fragmentan dos veces, una antes de GRE y otra después de IPv4sec.

El paquete se fragmenta antes de la encapsulación GRE y uno de estos paquetes GRE se fragmenta de nuevo después del cifrado IPv4sec.

Configurar "ip mtu 1440" (modo de transporte IPv4sec) o "ip mtu 1420" (modo de túnel IPv4sec) en el túnel GRE quitará la posibilidad de la fragmentación doble en esta situación.



1. El router recibe un datagrama de 1500 bytes.
2. Antes del encapsulamiento, el GRE fragmenta el paquete de 1500 bytes en dos partes: 1476 bytes (1500 - 24 = 1476) y 44 bytes (24 bytes de datos + encabezado IPv4 de 20 bytes) .
3. El GRE encapsula los fragmentos de IPv4, lo que agrega 24 bytes a cada paquete. Esto se traduce en dos paquetes de GRE + IPv4sec de 1500 bytes (1476 + 24 = 1500) y 68 bytes (44 + 24) cada uno.
4. IPv4sec cifra los dos paquetes, que añaden 52 bytes (modo de túnel IPv4sec) de sobrecarga de encapsulación a cada uno, para proporcionar un paquete de 1552 bytes y uno de 120 bytes.
5. El router fragmenta el paquete de IPv4sec de 1552 bytes, ya que es superior a la MTU

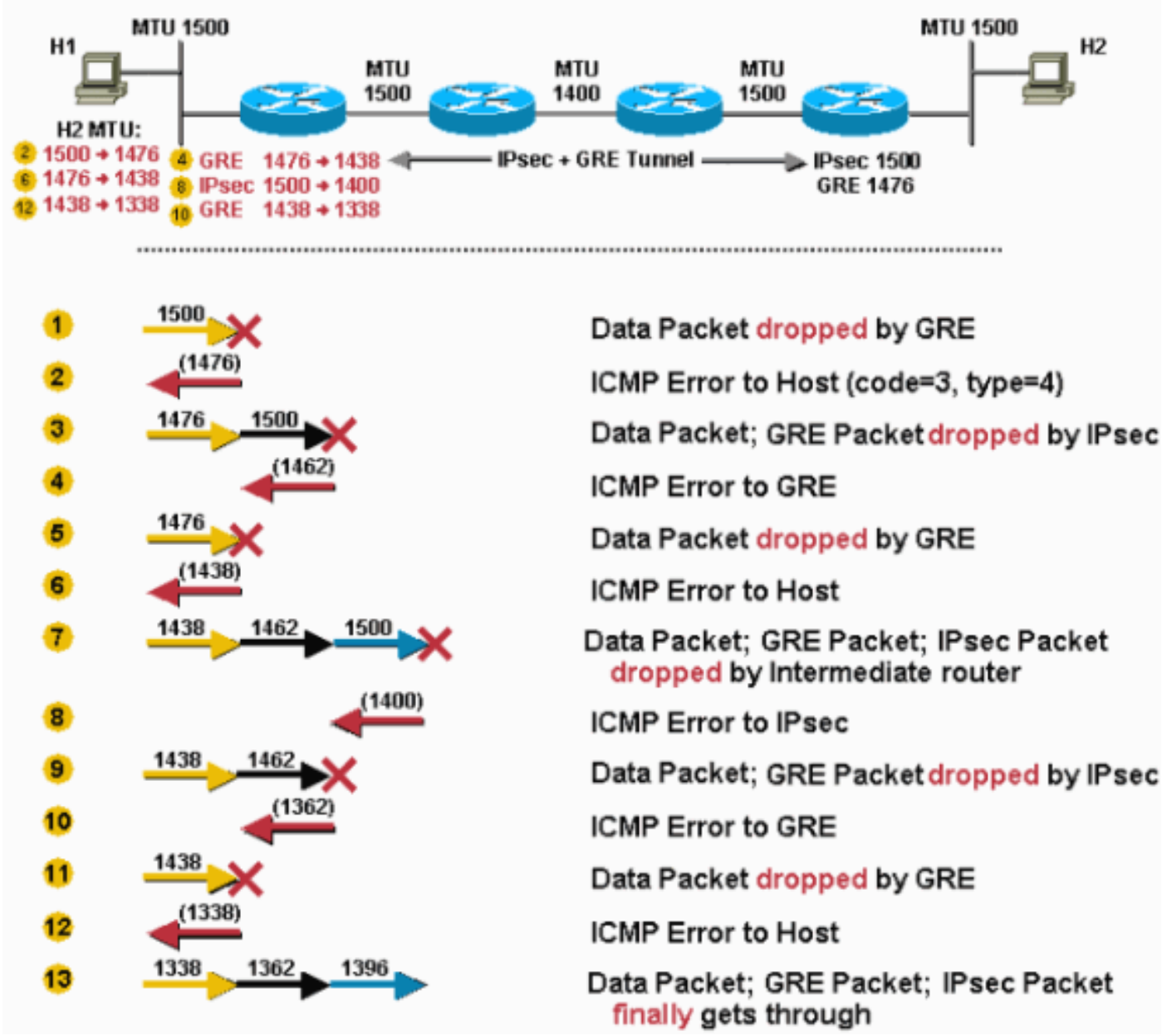
saliente (1500). El paquete de 1552 bytes se divide en partes, un paquete de 1500 bytes y un paquete de 72 bytes ("carga útil" de 52 bytes más encabezado IPv4 de 20 bytes adicional para el segundo fragmento). Los tres paquetes de 1500 bytes, 72 bytes y 120 bytes se dirigirán al par IPv4sec + GRE.

6. El router receptor reensambla los dos fragmentos de IPv4sec (1500 bytes y 72 bytes) para obtener el paquete original de IPv4sec + GRE de 1552 bytes. No debe hacerse nada en el paquete de IPv4sec + GRE de 120 bytes.
7. IPv4sec descifra los dos paquetes IPv4sec + GRE de 1552 bytes y 120 bytes para conseguir paquetes de GRE de 1500 bytes y 68 bytes.
8. El GRE desencapsula los dos paquetes de GRE de 1500 bytes y 68 bytes para conseguir fragmentos de paquetes de IPv4 de 1476 bytes y 44 bytes. Estos fragmentos de IPv4 se desviarán a un host de destino.
9. El host 2 reensambla estos fragmentos de IPv4 para obtener el datagrama IPv4 original de 1500 bytes.

La situación 10 es similar a la situación 8, salvo que hay un link de MTU inferior en la trayectoria de túnel. Esta es la peor situación para el primer paquete enviado del host 1 al host 2. Después del último paso en este escenario, el Host 1 establece la PMTU correcta para el Host 2 y todo funciona bien para las conexiones TCP entre el Host 1 y el Host 2. Los flujos TCP entre el host 1 y otros hosts (accesibles a través del túnel IPv4sec + GRE) solo tienen que pasar por los últimos tres pasos de la situación 10.

En este escenario, el **tunnel path-mtu-discovery** se configura en el túnel GRE y el bit DF se configura en los paquetes TCP/IPv4 que se originan desde el Host 1.

Ejemplo 10



- El router recibe un paquete de 1500 bytes. GRE descarta este paquete porque no puede fragmentar ni reenviar el paquete, ya que el bit DF está configurado y el tamaño del paquete excede la "MTU IP" de interfaz saliente una vez que se agrega la sobrecarga de GRE (24 bytes).
- El router envía un mensaje ICMP al host 1 para avisarle que la MTU del siguiente salto es de 1476 (1500 - 24 = 1476).
- El Host 1 cambia su PMTU para el Host 2 a 1476 y envía el tamaño más pequeño cuando retransmite el paquete. El GRE los encapsula y entrega el paquete de 1500 bytes a IPv4sec. IPv4sec descarta el paquete porque GRE ha copiado el bit DF (configurado) en el encabezado IPv4 interno con la sobrecarga de IPv4sec (máximo de 38 bytes); el paquete es demasiado grande para desviarlo de la interfaz física.
- IPv4sec envía un mensaje ICMP al GRE que indica que la MTU del siguiente salto es de 1462 bytes (ya que se agrega un máximo de 38 bytes para el cifrado y la sobrecarga de IPv4). GRE registra el valor 1438 (1462 - 24) como la "MTU IP" en la interfaz de túnel.

- Nota: Este cambio en el valor se almacena internamente y no se puede ver en la salida de la **show ip interface tunnel**<#>comando. Solo verá este cambio si a su vez utiliza el **debug tunnel** comando.

- La próxima vez que el Host 1 retransmita el paquete de 1476 bytes, GRE lo descartará.
- El router envía un mensaje ICMP al host 1 que indica que la MTU del siguiente salto es de 1438.
- El Host 1 disminuye la PMTU para el Host 2 y retransmite un paquete de 1438 bytes. Esta vez, el GRE acepta el paquete, lo encapsula y lo entrega a IPv4sec para el cifrado.
- El paquete de IPv4sec se reenvía al router intermedio y se descarta porque presenta una MTU de interfaz de salida de 1400.
- El router intermedio envía un mensaje de ICMP a IPv4sec en el que le avisa que la MTU del siguiente salto es de 1400. IPv4sec registra este valor en el valor de la PMTU de la SA de IPv4sec.
- Cuando el host 1 retransmite el paquete de 1438 bytes, el GRE lo encapsula y lo entrega a IPv4sec. IPv4sec descarta el paquete debido a que ha cambiado su propia PMTU a 1400.
- IPv4sec envía un error de ICMP a GRE que indica que la MTU del siguiente salto es de 1362 y el GRE registra el valor 1338 internamente.
- Cuando el Host 1 retransmite el paquete original (porque no recibió reconocimiento), GRE lo descarta.
- El router envía un mensaje ICMP al host 1 que indica que la MTU del siguiente salto es de 1338 (1362 - 24 bytes). El Host 1 disminuye su PMTU para el Host 2 a 1338.
- El Host 1 retransmite un paquete de 1338 bytes y esta vez puede pasarlo finalmente a través del Host 2.

Más Recomendaciones

Configuración del **tunnel path-mtu-discovery** en una interfaz de túnel puede ayudar a la interacción de GRE e IPv4sec cuando se configuran en el mismo router.

Sin el **tunnel path-mtu-discovery** , el bit DF siempre se borraría en el encabezado IPv4 de GRE.

Esto permite que el paquete de IPv4 del GRE se fragmente aunque el encabezado IPv4 de datos encapsulado tenga el bit DF configurado, lo que normalmente no permite la fragmentación del paquete.

Si **tunnel path-mtu-discovery** se configura en la interfaz de túnel GRE:

1. GRE copia el bit DF del encabezado IPv4 de datos al encabezado IPv4 de GRE.
2. Si el bit DF está configurado en el encabezado IPv4 de GRE y el paquete es "demasiado grande" después del cifrado de IPv4sec para la MTU de IPv4 en la interfaz física saliente, IPv4sec descarta el paquete y notifica al túnel GRE que reduzca su tamaño de MTU de IPv4.
3. IPv4sec realiza la PMTUD para sus propios paquetes y, si la PMTU de IPv4sec cambia (si se reduce), IPv4sec no notifica inmediatamente al GRE, pero cuando otro paquete más grande pasa, se produce el proceso del paso 2.

4. La MTU IPv4 de GRE es ahora más pequeña, por lo que descarta cualquier paquete IPv4 de datos con el conjunto de bits DF que ahora sea demasiado grande y envía un mensaje ICMP al host de envío.

tunnel path-mtu-discovery ayuda a la interfaz GRE a establecer su MTU IPv4 dinámicamente, en lugar de estáticamente con el comando **ip mtu** comando. En realidad, se recomienda utilizar ambos comandos.

ip mtu se utiliza para proporcionar espacio para el GRE y la sobrecarga de IPv4sec en relación con la MTU de IPv4 de la interfaz física saliente local.

tunnel path-mtu-discovery permite que la MTU de IPv4 del túnel GRE se reduzca aún más si hay un link de MTU de IPv4 más bajo en la trayectoria entre los peers de IPv4sec.

Estas son algunas de las cosas que puede hacer si tiene problemas con la PMTUD en una red donde hay túneles GRE + IPv4sec configurados.

Esta lista comienza con la solución preferida.

1. Solucione el problema que impide el funcionamiento de PMTUD, que suele tener su origen en un router o firewall que bloquea ICMP.
2. Use el comando **ip tcp adjust-mss** en las interfaces de túnel para que el router reduzca el valor de MSS de TCP en el paquete SYN de TCP. Esto ayuda a los dos hosts finales (el emisor y el receptor TCP) a utilizar paquetes lo suficientemente pequeños como para que no se necesite PMTUD.
3. Utilice el routing de políticas en la interfaz de entrada del router y configure un mapa de ruta para borrar el bit DF en el encabezado IPv4 de datos antes de que llegue a la interfaz de túnel GRE. Esto permite que el paquete de IPv4 de datos se fragmente antes de la encapsulación GRE.
4. Aumente la "MTU IP" en la interfaz de túnel GRE para que sea igual a la MTU de interfaz saliente. Esto permite que el paquete de IPv4 de datos sea encapsulado GRE sin fragmentarlo primero. El paquete GRE se cifra en IPv4sec y, a continuación, se fragmenta para salir de la interfaz de salida física. En este caso, no configuraría **tunnel path-mtu-discovery** en la interfaz de túnel GRE. Esto puede reducir drásticamente el rendimiento debido a que el montaje del paquete de IPv4 en el par IPv4sec se realiza en el modo de cambio de proceso.

Información Relacionada

- Página de Soporte de IP Routing
- Página de Soporte de IPSec (IP Security Protocol)
- RFC 1191:Detección de MTU de Trayectoria
- RFC 1063 Opciones de Detección de MTU IP
- RFC 791 Internet Protocol
- RFC 793 Protocolo de Control de Transmisión
- RFC 879 Tamaño Máximo de Segmento de TCP y Temas Relacionados
- RFC 1701 Generic Routing Encapsulation (GRE)
- RFC 1241 Esquema de un Protocolo de Encapsulación de Internet

- RFC 2003 RFC 2003 Encapsulación de IP dentro de IP
- Soporte Técnico y Documentación - Cisco Systems

Acerca de esta traducción
Cisco ha traducido este documento combinando la traducción automática y los recursos humanos a fin de ofrecer a nuestros usuarios en todo el mundo contenido en su propio idioma.

Tenga en cuenta que incluso la mejor traducción automática podría no ser tan precisa como la proporcionada por un traductor profesional.

Cisco Systems, Inc. no asume ninguna responsabilidad por la precisión de estas traducciones y recomienda remitirse siempre al documento original escrito en inglés (insertar vínculo URL).