Resolver problemas de fragmentação de IPv4, MTU, MSS e PMTUD com GRE e IPsec

Contents

Introdução

Este documento descreve como a fragmentação IPv4 e a descoberta de unidade de transmissão máxima de caminho (PMTUD) funcionam.

Informações de Apoio

Também são discutidos os cenários que envolvem o comportamento de PMTUD em conjunto com diferentes combinações de túneis IPv4.

Fragmentação e remontagem de IPv4

Embora o comprimento máximo de um datagrama IPv4 seja 65535, a maioria dos links de transmissão impõe um limite menor de comprimento máximo do pacote, chamado MTU. O valor de MTU depende do link de transmissão.

O design do IPv4 comporta diferenças de MTU, pois permite aos roteadores fragmentarem datagramas IPv4 conforme a necessidade.

A estação receptora é responsável pela remontagem dos fragmentos no datagrama IPv4 original de tamanho completo.

A fragmentação de IPv4 divide um datagrama em partes que são remontadas posteriormente.

A origem, o destino, a identificação, o tamanho total e os campos de deslocamento do fragmento de IPv4, juntamente com os sinalizadores de "mais fragmentos" (MF) e "não fragmentar" (DF) no cabeçalho do IPv4, são usados para fragmentação e remontagem de IPv4.

Para obter mais informações sobre a mecânica da fragmentação de IPv4 e da remontagem, consulte RFC 791

Esta imagem mostra o layout de um cabeçalho 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

A identificação é de 16 bits e é um valor atribuído pelo remetente de um datagrama IPv4. Isso ajuda na remontagem dos fragmentos de um datagrama.

O deslocamento do fragmento é de 13 bits e indica onde um fragmento pertence no datagrama de IPv4 original. Esse valor é um múltiplo de 8 bytes.

Existem 3 bits para sinalizadores de controle no campo de sinalizadores do cabeçalho IPv4. O bit "não fragmentar" (DF) determina se um pacote pode ser fragmentado.

O Bit 0 é reservado e sempre definido como 0.

O Bit 1 é o bit DF (0 = "pode fragmentar", 1 = "não fragmentar").

O Bit 2 é o bit MF (0 = "último fragmento", 1 = "mais fragmentos").

Valor	Bit 0 reservado	Bit 1 DF	Bit 2 MF
0	0	Maio	Sobrenome
1	0	Errado	Mais

Se os comprimentos dos fragmentos IPv4 forem adicionados, o valor excederá em 60 o comprimento do datagrama IPv4 original.

O motivo pelo qual o comprimento total é aumentado em 60 é porque três cabeçalhos IPv4 adicionais foram criados, um para cada fragmento após o primeiro fragmento.

O primeiro fragmento tem um desvio de 0. O tamanho desse fragmento é 1500. Isso inclui 20 bytes para o cabeçalho IPv4 original ligeiramente modificado.

O segundo fragmento tem um deslocamento de 185 (185 x 8 = 1.480). A parte de dados desse fragmento inicia em 1.480 bytes no datagrama IPv4 original.

O comprimento desse fragmento é 1.500. Isso inclui o cabeçalho IPv4 adicional criado para esse fragmento.

O terceiro fragmento tem um deslocamento de 370 (370 x 8 = 2.960). A parte de dados desse fragmento inicia em 2.960 bytes no datagrama IPv4 original.

O comprimento desse fragmento é 1.500. Isso inclui o cabeçalho IPv4 adicional criado para esse fragmento.

O quarto fragmento tem um deslocamento de 555 (555 x 8 = 4.440), o que significa que a parte de dados desse fragmento inicia em 4.440 bytes no datagrama IPv4 original.

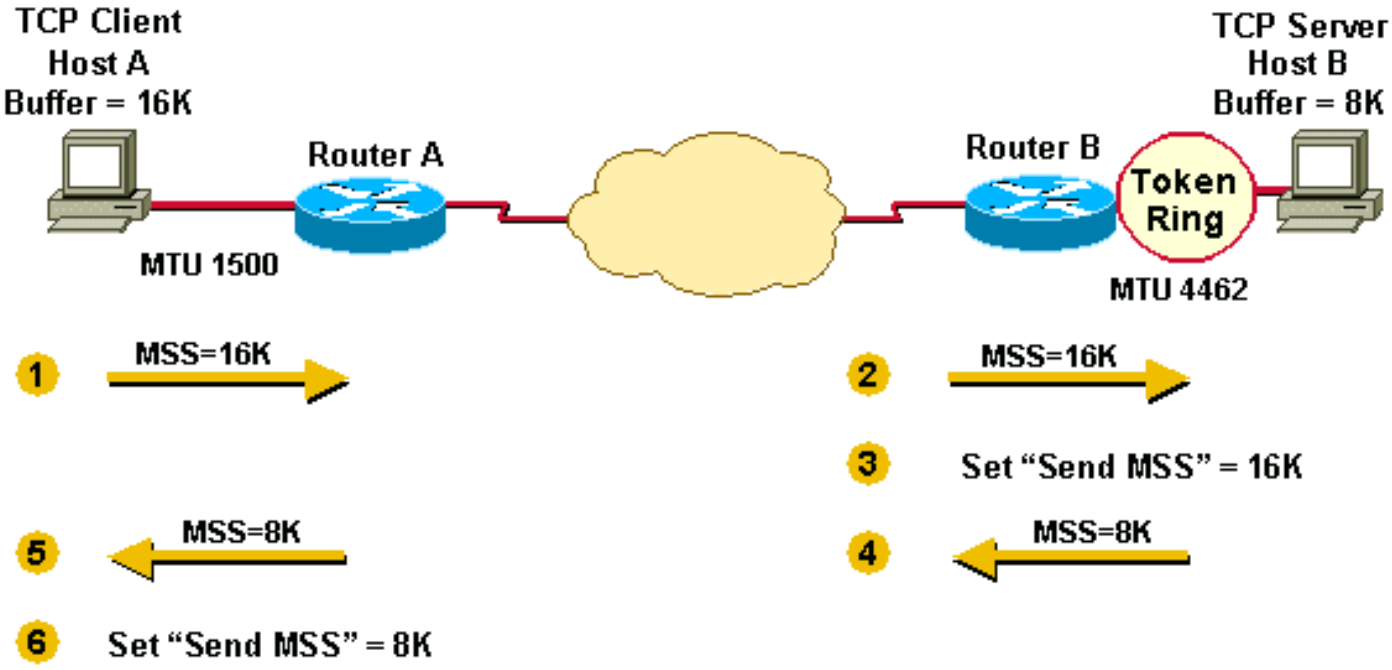
O comprimento desse fragmento é 700. Isso inclui o cabeçalho IPv4 adicional criado para esse fragmento.

O tamanho do datagrama IPv4 original só pode ser determinado quando o último fragmento for recebido.

O deslocamento do fragmento no último fragmento (555) fornece um deslocamento de dados de 4.440 bytes para o datagrama IPv4 original.

A soma dos bytes de dados do último fragmento (680 = 700 - 20) produz 5.120 bytes, que é a porção de dados do datagrama IPv4 original.

A adição de 20 bytes para um cabeçalho IPv4 é igual ao tamanho do datagrama IPv4 original (4.440 + 680 + 20 = 5140), conforme mostrado nas imagens.



Problemas com fragmentação de IPv4

A fragmentação de IPv4 resulta em um pequeno aumento na sobrecarga de CPU e memória para fragmentar um datagrama de IPv4. Isso se aplica ao remetente e a um roteador no caminho entre um remetente e um destinatário.

A criação de fragmentos envolve a criação de cabeçalhos de fragmento e copia o datagrama original nos fragmentos.

Isso é feito de forma eficiente, porque as informações necessárias para criar os fragmentos estão imediatamente disponíveis.

A fragmentação provoca mais "overhead" para o destinatário durante a remontagem dos fragmentos, porque o destinatário deve alocar memória para os fragmentos de chegada e agrupá-los novamente em um datagrama, depois do recebimento de todos os fragmentos.

A remontagem em um host não é considerada um problema, porque o host possui os recursos de memória e o tempo para se dedicar a essa tarefa.

No entanto, a remontagem é ineficiente em um roteador, cuja principal função é encaminhar pacotes o mais rápido possível.

Um roteador não é projetado para reter os pacotes por qualquer período.

Um roteador que faz a remontagem escolhe o maior buffer disponível (18k), porque ele só determinará o tamanho do pacote IPv4 original quando o último fragmento for recebido.

Outro problema da fragmentação envolve o modo de manuseio dos fragmentos descartados.

Se um fragmento de um datagrama IPv4 for descartado, então todo o datagrama IPv4 original deverá estar presente e será fragmentado.

Isso é observado com o sistema de arquivos de rede (NFS). O NFS tem um tamanho de bloco de leitura e gravação de 8192.

Portanto, um datagrama IPv4/UDP de NFS é de aproximadamente 8.500 bytes (incluindo cabeçalhos NFS, UDP e IPv4).

Uma estação de envio, ligada a uma Ethernet (MTU 1500), deve fragmentar o datagrama de 8.500 bytes em seis (6) pedaços, cinco (5) fragmentos de 1.500 bytes e um (1) fragmento de 1.100 bytes.

Se qualquer um dos seis fragmentos for descartado por causa de um link congestionado, o datagrama original completo precisará ser retransmitido. Isso resulta em mais seis fragmentos a serem criados.

Caso esse link descarte um a cada seis pacotes, é baixa a probabilidade de transferência de quaisquer dados de NFS, pois pelo menos um fragmento IPv4 seria descartado em cada datagrama IPv4 original de 8.500 bytes de NFS.

Os firewalls que filtram ou manipulam pacotes de informações da Camada 4 (L4) a Camada 7 (L7) têm dificuldade em processar fragmentos IPv4 corretamente.

Se os fragmentos IPv4 estiverem fora de ordem, um firewall bloqueará os fragmentos não iniciais, pois não carregam as informações que correspondem ao filtro de pacote.

Isso significa que o datagrama IPv4 original não pode ser recomposto pelo host de recebimento.

Se o firewall está configurado para permitir fragmentos não iniciais com informações insuficientes para corresponder corretamente ao filtro, é possível ocorrer um ataque de fragmento não inicial pelo firewall.

Os dispositivos de rede, como Mecanismos de Switch de Conteúdo, direcionam os pacotes de acordo com as informações de L4 a L7, e se um pacote abrange vários fragmentos, o dispositivo tem dificuldade em aplicar as políticas.

Evite a fragmentação de IPv4: como funciona o TCP MSS

O tamanho máximo de segmento (MSS) do protocolo TCP define o volume máximo de dados que um host aceita em um único conjunto de dados TCP/IPv4.

Este datagrama de TCP/IPv4 possivelmente é fragmentado na camada de IPv4. O valor MSS é enviado como uma opção de cabeçalho TCP somente em segmentos TCP SYN.

Cada lado de uma conexão TCP relata seu valor MSS para o outro lado. O valor MSS não é negociado entre os hosts.

O host remetente é necessário para limitar o tamanho dos dados em um único segmento TCP para um valor menor ou igual ao MSS relatado pelo host destinatário.

Originalmente, o MSS mostrava o tamanho de um buffer (maior ou igual a 65.496 bytes) alocado

em uma estação receptora para armazenar os dados TCP contidos em um único datagrama IPv4.

O MSS foi o máximo segmento de dados que o receptor TCP estava disposto a aceitar. Esse segmento TCP poderia ser tão grande quanto 64K e fragmentado na camada IPv4 para transmissão para o host de destino.

O host receptor remontaria o datagrama de IPv4 antes de ele enviar o segmento TCP completo para a camada de TCP.

Como os valores de MSS são definidos e usados para limitar o segmento TCP e os tamanhos de datagrama IPv4.

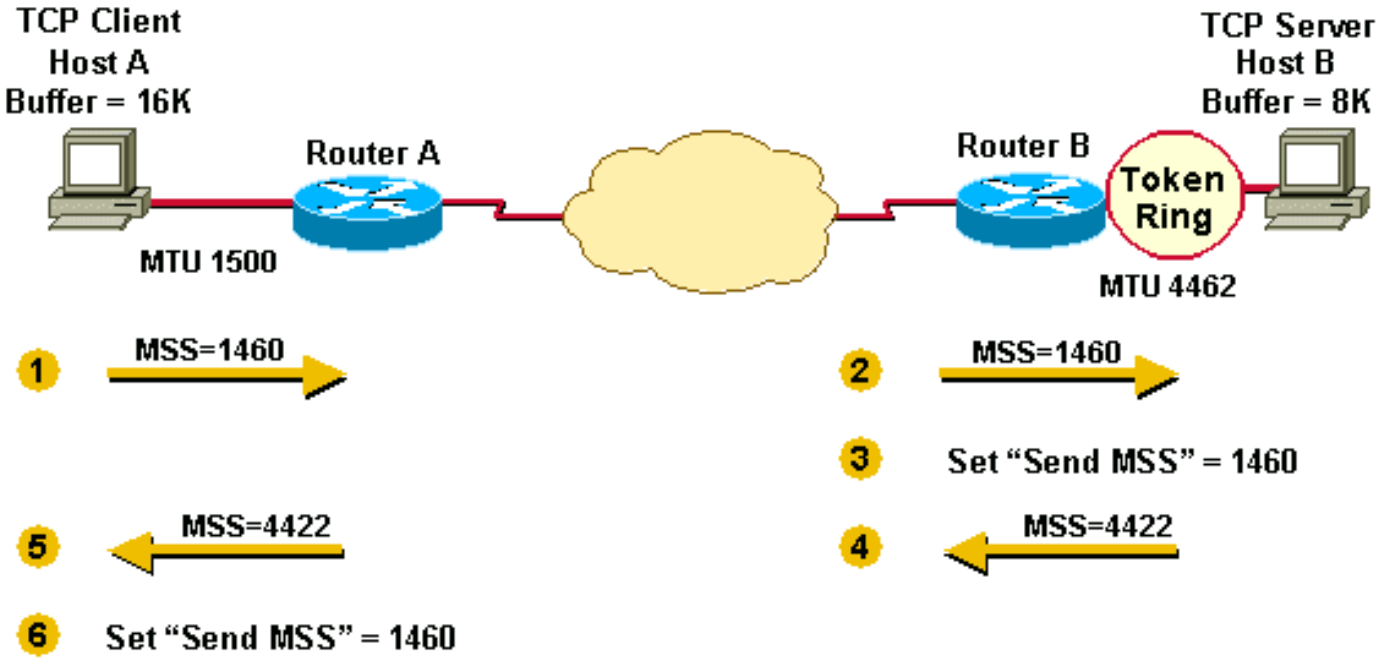
O exemplo 1 ilustra a maneira que o MSS foi implementado pela primeira vez.

O Host A tem um buffer de 16K e o Host B um buffer de 8K. Eles enviam e recebem seus valores de MSS e ajustam o MSS de envio para enviar dados um ao outro.

O Host A e o Host B precisam fragmentar os datagramas IPv4 maiores do que a interface MTU, porém menores que o MSS enviado, porque a pilha TCP transmite 16K ou 8K bytes de dados para a pilha de IPv4.

No caso do Host B, os pacotes são fragmentados para chegar à LAN de Token Ring e, novamente, para chegar à LAN de Ethernet.

Exemplo 1



1. O Host A envia o valor MSS de 16K para o Host B.
2. O Host B recebe o valor MSS de 16K do Host A.
3. O Host B define o valor MSS de envio como 16K.
4. O Host B envia o valor MSS de 8K para o Host A.
5. O Host A recebe o valor MSS de 8K do Host B.

6. O Host A define o valor MSS de envio como 8K.

Para ajudar a evitar fragmentação de IPv4 nos endpoints da conexão de TCP, a seleção do valor MSS foi trocada para o tamanho de buffer mínimo e para o MTU da interface de saída (- 40).

Os números de MSS são menores que os números de MTU de 40 bytes porque o MSS (o tamanho de dados de TCP) não inclui o cabeçalho IPv4 de 20 bytes e o cabeçalho TCP de 20 bytes.

O MSS baseia-se nos tamanhos de cabeçalho padrão. A pilha de remetente deve subtrair os valores apropriados para o cabeçalho IPv4, e o cabeçalho TCP depende de quais opções de IPv4 ou TCP são usadas.

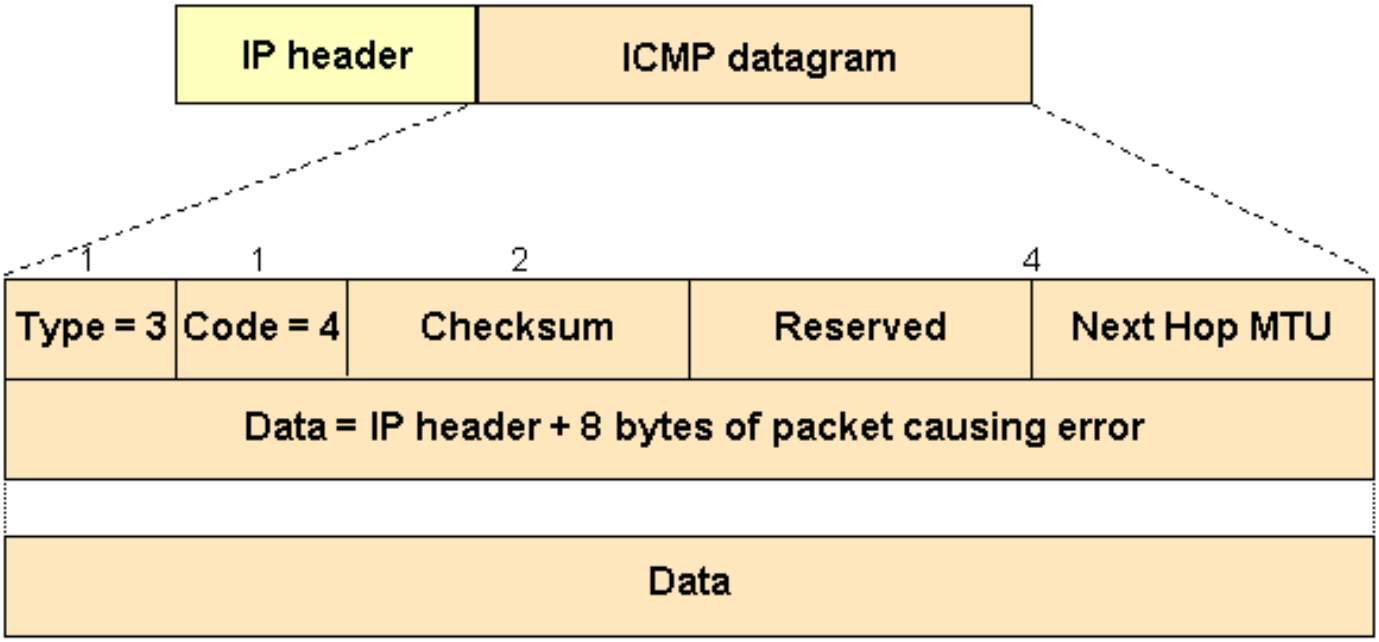
No momento, o MSS funciona de uma maneira que cada host compara primeiro o MTU da interface de saída com seu próprio buffer e escolhe o valor mais baixo como o MSS a ser enviado.

Em seguida, os hosts comparam o tamanho do MSS recebido com seu próprio MTU de interface e escolhem novamente o menor dos dois valores.

O exemplo 2 ilustra esta etapa adicional executada pelo remetente, a fim de evitar a fragmentação nos fios locais e remotos.

O MTU da interface de saída é considerado por cada host, antes dos host trocarem entre si os valores de MSS. Isso ajuda a evitar a fragmentação.

Exemplo 2



1. O Host A compara o buffer de MSS (16K) e seu MTU (1.500 - 40 = 1.460) e usa o valor mais baixo como MSS (1.460) para envio ao Host B.
2. O Host B recebe o MSS de envio (1.460) do Host A e compara com o valor de sua interface

de saída de MTU - 40 (4.422).
3. O Host B define o valor mais baixo (1.460) como o MSS para enviar datagramas IPv4 para o Host A.
4. O Host B compara o buffer de MSS (8K) e o MTU (4.462-40 = 4.422) e usa 4.422 como MSS para enviar ao Host A.
5. O Host A recebe o MSS de envio (4.422) do Host B e compara com o valor de sua interface de saída de MTU -40 (1.460).
6. O Host A define o valor mais baixo (1.460) como o MSS para enviar datagramas IPv4 para o Host B.

1460 é o valor escolhido por ambos os hosts como o MSS de envio de um para o outro. Muitas vezes, o valor do MSS de envio é o mesmo em cada extremidade de uma conexão TCP.

No exemplo 2, a fragmentação não ocorre nos endpoints de uma conexão TCP, porque os hosts consideram os dois MTUs de interface de saída.

Os pacotes ainda ficarão fragmentados na rede entre o Roteador A e o Roteador B, se encontrarem um link com um MTU mais baixo do que a da interface de saída de qualquer um dos hosts.

O que é PMTUD

O MSS de TCP aborda a fragmentação em dois endpoints de uma conexão TCP, mas não interfere quando há um link de MTU menor no meio, entre esses dois endpoints.

O PMTUD foi desenvolvido para evitar a fragmentação no caminho entre os endpoints. Ele é usado para determinar dinamicamente o MTU mais baixo ao longo do caminho, da origem de um pacote até o destino.

Observação: o PMTUD é compatível somente com TCP e UDP. Outros protocolos não são compatíveis. Se o PMTUD for ativado em um host, todos os pacotes TCP e UDP provenientes do host terão o bit DF definido.

Quando um host envia um pacote de dados MSS completo com o conjunto de bits DF, o PMTUD reduz o valor do MSS de envio para a conexão, caso ele receba a informação de que o pacote precisa de fragmentação.

Um host registra o valor de MTU para um destino, pois ele cria uma entrada de host (/32) em sua tabela de roteamento com esse valor de MTU.

Se um roteador tenta encaminhar um datagrama IPv4 (com o conjunto de bits DF) em um link que tem um MTU menor que o tamanho do pacote, o roteador descarta o pacote e retorna uma mensagem "Destination Unreachable" do Internet Control Message Protocol (ICMP) para a origem do datagrama IPv4, com o código que indica "fragmentation needed and DF set" (tipo 3, código 4).

Quando a estação de origem recebe a mensagem ICMP, diminui o MSS de envio e, quando o

TCP retransmite o segmento, usa o tamanho menor do segmento.

Aqui está um exemplo de uma mensagem ICMP "fragmentation needed and DF set" vista em um roteador depois que o **debug ip icmp** comando é ativado:

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

Este diagrama mostra o formato do cabeçalho de ICMP de uma mensagem "fragmentação necessária e DF definido" "Destino inacessível".

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

De acordo com RFC 1191, um roteador que retorna uma mensagem ICMP indicando "fragmentação necessária e DF definido" deve incluir o MTU da rede de próximo salto nos 16 bits de ordem inferior do campo de cabeçalho adicional de ICMP, rotulado como "não utilizado" na especificação do ICMP RFC 792.

As implementações iniciais do RFC 1191 não forneceram as informações de MTU do próximo salto. Mesmo quando essa informação tiver sido fornecida, alguns hosts vão ignorá-la.

Para esse caso, o RFC 1191 também contém uma tabela que lista os valores sugeridos que são usados para diminuir o MTU durante o PMTUD.

Ele é usado por hosts para chegar com mais rapidez a um valor razoável para o MSS de envio conforme mostrado neste exemplo.

O PMTUD é realizado sempre em todos os pacotes, pois o caminho entre o remetente e o destinatário pode mudar dinamicamente.

Sempre que um remetente receber mensagens "Cannot Fragment" do ICMP, ele atualizará as informações de roteamento (onde armazena o PMTUD).

Duas coisas podem acontecer durante o PMTUD:

1. O pacote pode chegar até o receptor sem ser fragmentado.

Observação: para que um roteador proteja a CPU contra ataques DoS, ele limita para duas pessoas por segundo o número de mensagens ICMP inacessíveis enviadas. Portanto, nesse contexto, se você tiver um cenário de rede no qual espera que o roteador precise responder com mais de duas mensagens ICMP (tipo = 3, código = 4) por segundo (podem ser hosts diferentes), desabilite o controle de fluxo de mensagens ICMP com o **no ip icmp rate-limit unreachable [df] interface** comando.

---

2. O remetente recebe mensagens "Cannot Fragment" do ICMP nos saltos ao longo do caminho para o destinatário.

O PMTUD é efetuado independentemente de ambas as direções de um fluxo de TCP.

Há casos em que o PMTUD em uma direção de fluxo aciona uma das estações finais para diminuir o MSS de envio e a outra estação final mantém o MSS de envio original, porque ela nunca enviou um datagrama IPv4 grande o suficiente para acionar o PMTUD.

Um exemplo é a conexão HTTP descrita no Exemplo 3. O cliente TCP envia pacotes pequenos e o servidor envia pacotes grandes.

Nesse caso, apenas os pacotes grandes do servidor (maiores de 576 bytes) acionam o PMTUD.

Os pacotes do cliente são pequenos (menos de 576 bytes) e não acionam o PMTUD porque não exigem fragmentação para passar pelo link de MTU 576.

Exemplo 3

O exemplo 4 mostra um exemplo de roteamento assimétrico, em que um dos caminhos tem um MTU mínimo menor que o outro.

O roteamento assimétrico ocorre quando diferentes caminhos são usados para enviar e receber dados entre os dois endpoints.

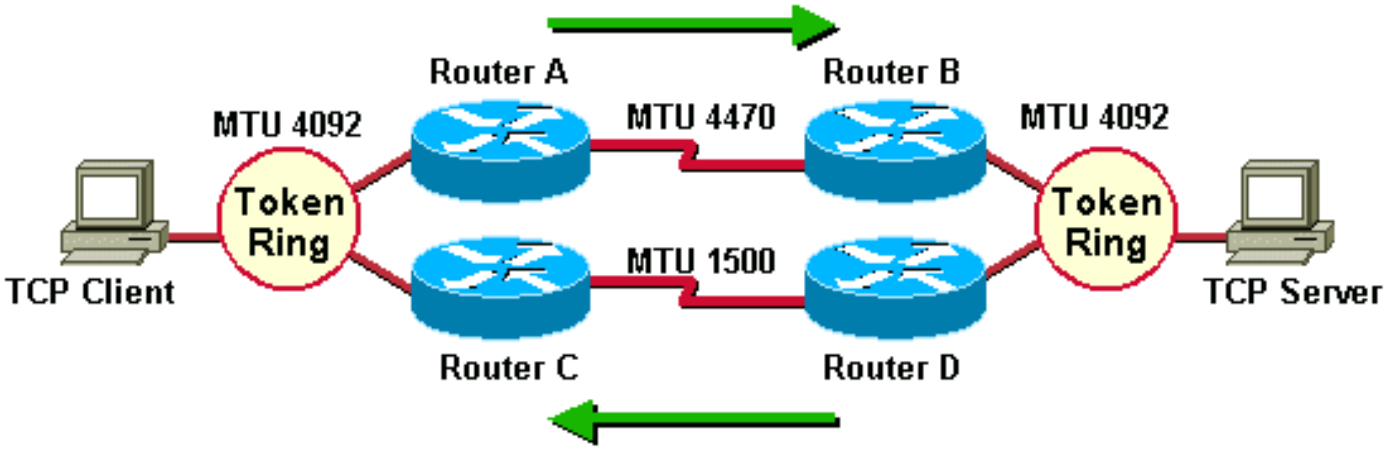
Nesse exemplo, o PMTUD aciona a redução do MSS de envio apenas em uma direção de um fluxo de TCP.

O tráfego do cliente de TCP para o servidor passa pelos Roteadores A e B, enquanto o tráfego de retorno proveniente do servidor para o cliente passa pelos Roteadores D e C.

Quando o servidor TCP envia pacotes para o cliente, o PMTUD aciona o servidor para reduzir o MSS de envio, pois o Roteador D deve fragmentar os pacotes de 4.092 bytes antes de enviá-los ao Roteador C.

Por outro lado, o cliente nunca recebe uma mensagem "Destination Unreachable" do ICMP com o código que indica "fragmentation needed and DF set", porque o Roteador A não precisa fragmentar os pacotes quando os envia ao servidor pelo Roteador B.

Exemplo 4



Observação: **o comando ip tcp path-mtu-discovery é usado para habilitar a descoberta de caminho TCP MTU para conexões TCP iniciadas por roteadores (BGP e Telnet, por exemplo).**

Problemas com o PMTUD

Essas são ações que podem interromper o PMTUD.

- Um roteador descarta um pacote e não envia uma mensagem ICMP. (Incomum)
- Um roteador gera e envia uma mensagem ICMP, mas a mensagem ICMP fica bloqueada por um roteador ou firewall entre esse roteador e o remetente. (Comum)
- Um roteador gera e envia uma mensagem ICMP que será ignorada pelo remetente. (Incomum)

O primeiro e o último dos três marcadores aqui são geralmente o resultado de um erro, mas o marcador central descreve um problema comum.

Os que implementam filtros de pacotes ICMP tendem a bloquear todos os tipos de mensagens ICMP, em vez de bloquear apenas determinados tipos de mensagens ICMP.

Um filtro de pacotes pode bloquear todos os tipos de mensagens ICMP, exceto as que contêm "unreachable" ou "time-exceeded".

O sucesso ou a falha do PMTUD depende das mensagens ICMP inacessíveis que chegam ao remetente de um pacote TCP/IPv4.

As mensagens ICMP de tempo excedido são importantes para outros problemas de IPv4.

Um exemplo desse filtro de pacotes, implementado em um roteador, é mostrado aqui.

```
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
```

Há outras técnicas que podem ser usadas para aliviar o problema de um ICMP totalmente bloqueado.

- Limpe o bit DF no roteador e permita a fragmentação. (No entanto, essa não é uma boa ideia. Consulte Problemas com a fragmentação de IP para obter mais informações).

- Manipule o valor da opção TCP MSS MSS com o comando de interface **ip tcp adjust-mss <500-1460>**.

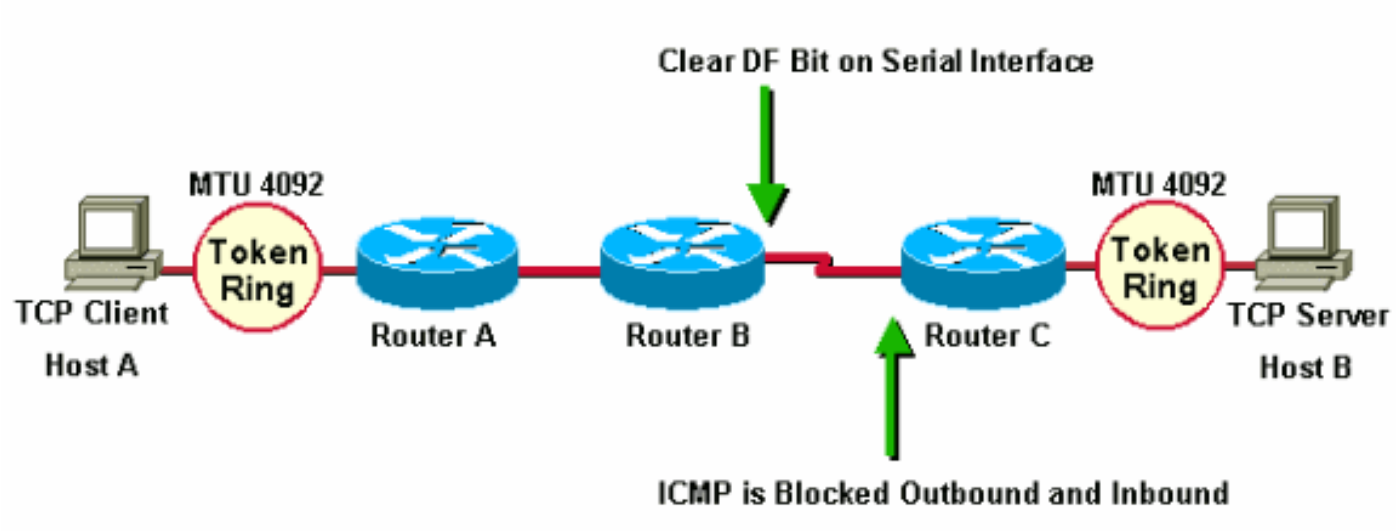
No próximo exemplo, os Roteadores A e B estão no mesmo domínio administrativo. O Roteador C é inacessível e bloqueia o ICMP, então o PMTUD é interrompido.

Uma solução para essa situação é limpar o bit DF em ambos sentidos no Roteador B, para habilitar a fragmentação. Isso pode ser feito com o roteamento de política.

A sintaxe para limpar o bit DF está disponível no Cisco IOS® Software Versão 12.1(6) e posterior.

```
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
```



Outra opção é alterar o valor de opção TCP MSS em pacotes SYN que atravessam o roteador (disponível no Cisco IOS® 12.2(4)T e posterior).

Isso reduz o valor da opção MSS no pacote TCP SYN para que seja menor que o valor (1460) no **ip tcp adjust-mss**comando.

O resultado é que o remetente TCP não envia segmentos maiores que esse valor.

O tamanho do pacote IPv4 é 40 bytes (1.500) maior do que o valor do MSS (1.460 bytes) para incluir o cabeçalho TCP (20 bytes) e o cabeçalho IPv4 (20 bytes).

Você pode ajustar o MSS dos pacotes TCP SYN com o **ip tcp adjust-mss** comando. Essa sintaxe reduz o valor de MSS nos segmentos de TCP para 1.460.

Esse comando afeta o tráfego de entrada e saída na interface serial0.

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

Os problemas de fragmentação de IPv4 tornaram-se mais conhecidos desde que as implantações de túneis IPv4 aumentaram.

Os túneis provocam mais fragmentação, pois o encapsulamento do túnel adiciona um “overhead" ao tamanho do pacote.

Por exemplo, a adição de encapsulamento de roteador genérico (GRE) aumenta um pacote em 24 bytes e, após esse aumento, o pacote precisa ser fragmentado porque é maior do que o MTU de saída.

Topologias de rede comuns que necessitam de PMTUD

O PMTUD é necessário nas situações de rede em que os links intermediários têm MTUs menores que o MTU dos links finais. Algumas razões comuns para a existência desses enlaces de MTU menores são:

-

Hosts terminais conectados por Token Ring (ou FDDI), com uma conexão Ethernet entre eles. Os MTUs de Token Ring (ou FDDI) nas extremidades são maiores que o MTU de Ethernet localizado no meio.

- O PPPoE (geralmente utilizado com ADSL) precisa de 8 bytes para seu cabeçalho. Isso reduz o MTU efetivo de Ethernet para 1492 (1500 - 8).

Os protocolos de túnel, como o GRE, IPv4sec, L2TP, também precisam de espaço para seus respectivos cabeçalhos e rodapés. Isso também reduz o MTU efetivo da interface de saída.

Túnel

Um túnel é uma interface lógica em um roteador da Cisco, proporcionando uma maneira de encapsular pacotes passageiros dentro de um protocolo de transporte.

É uma arquitetura projetada para prestar serviços para implementação em um esquema de encapsulamento ponto a ponto. As interfaces de túnel têm estes três componentes principais:

- Protocolo de passageiro (AppleTalk, Banyan VINES, CLNS, DECnet, IPv4 ou IPX)
- Protocolo de portador - um desses protocolos de encapsulamento:
  - GRE – Protocolo de portador de multiprotocolo da Cisco. Consulte RFC 2784 eRFC 1701 para obter mais informações.
  - IPv4 em túneis de IPv4 - Consulte RFC 2003 para obter mais informações.
- Protocolo de transporte - O protocolo usado para transportar o protocolo encapsulado.

Os pacotes mostrados nesta seção ilustram os conceitos de tunelamento de IPv4, em que o GRE é o protocolo de encapsulamento e o IPv4 é o protocolo de transporte.

O protocolo de passageiro também é IPv4. Nesse caso, o IPv4 é o protocolo de transporte e de passageiros.

Pacote normal

IPv4	TCP	Telnet

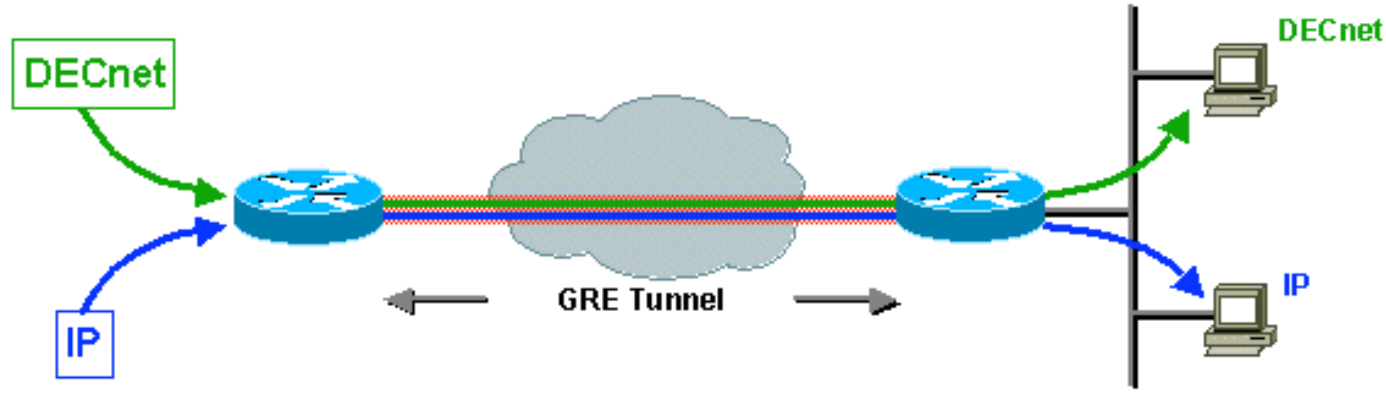
Pacote de túneis

IPv4	GRE	IPv4	TCP	Telnet

- IPv4 é o protocolo de transporte
- GRE é o protocolo de encapsulamento.
- IPv4 é o protocolo de passageiros.

O próximo exemplo mostra o encapsulamento de IPv4 e DECnet como protocolos de passageiro com GRE como transportador.

Isso ilustra a possibilidade de que os protocolos do portador encapsulem vários protocolos de passageiro, conforme mostrado na imagem.



Um administrador de rede considera um tunelamento quando há duas redes não contíguas e sem IPv4 separadas por um backbone de IPv4.

Caso as redes não contíguas executem o DECnet, o administrador pode optar por conectá-las entre si (ou não) ao configurar o DECnet no backbone.

O administrador não vai querer permitir que o roteamento de DECnet consuma a largura de banda do backbone, porque isso interferiria no desempenho da rede IPv4.

Uma alternativa viável é fazer um túnel DECnet no backbone IPv4. A solução de túnel encapsula os pacotes DECnet dentro do IPv4 e os envia pelo backbone até o endpoint de túnel, onde o encapsulamento é removido e os pacotes DECnet são encaminhados ao seu destino através de DECnet.

Existem vantagens ao encapsular o tráfego dentro de um outro protocolo:

- Os endpoints usam endereços privados (RFC 1918) e o backbone não oferece suporte ao roteamento desses endereços.
- Permitem redes virtuais privadas (VPNs) através de WANs ou Internet.
- Unem redes de multiprotocolos descontíguas em um backbone de protocolo único.
- Criptografam o tráfego ao longo do backbone ou da Internet.

A partir deste ponto, o IPv4 é usado como protocolo de passageiro e protocolo de transporte.

Considerações com relação às interfaces de túnel

Estas são considerações sobre tunelamento.

- O switching rápido de túneis GRE foi introduzido no Cisco IOS® versão 11.1 e o switching de CEF foi introduzido na versão 12.0.
- O switching de CEF para túneis GRE de vários pontos foi introduzido na versão 12.2(8)T.
- O encapsulamento e o desencapsulamento no terminal de túnel eram operações lentas em versões anteriores do Cisco IOS® , quando havia suporte somente ao switching de processo.

- Há problemas de segurança e topologia no tunelamento de pacotes. Os túneis podem desviar listas de controle de acesso (ACLs) e firewalls.

- Se você criar um túnel através de um firewall, vai ignorar o protocolo do passageiro que está sendo encapsulado. Portanto, é recomendável incluir a funcionalidade do firewall nos endpoints do túnel para aplicar qualquer política aos protocolos de passageiro.

- O tunelamento cria problemas com os protocolos de transporte com temporizadores limitados (por exemplo, DECnet) por causa do aumento da latência.

- O tunelamento em ambientes com links de velocidade diferentes, como anéis FDDI rápidos e através de linhas telefônicas lentas de 9.600 bps, apresenta problemas de reordenação de pacotes. Alguns protocolos passageiros apresentam baixo desempenho nas redes de mídia mista.

- Os túneis de ponto a ponto consomem largura de banda em um link físico. Em vários túneis de ponto a ponto, cada interface de túnel tem uma largura de banda e a interface física na qual o túnel passa tem uma largura de banda. Por exemplo, defina a largura de banda do túnel como 100 Kb, se houvesse 100 túneis passando por um link de 10 Mb. A largura de banda padrão para um túnel é 9Kb.

- Protocolos de roteamento preferem um túnel que passa por um link real, porque o túnel parece um link de um salto com o caminho de custo mais baixo, embora envolva mais saltos e, portanto, tenha maior custo que outro caminho. Isso é mitigado com a configuração adequada do protocolo de roteamento. Execute um protocolo de roteamento diferente na interface do túnel em relação ao protocolo de roteamento em execução na interface física.

- Os problemas com o roteamento recursivo são evitados ao configurar as rotas estáticas apropriadas para o destino do túnel. Uma rota recursiva ocorre quando o melhor caminho para o destino do túnel é pelo próprio túnel. Essa situação faz com que a interface de túnel salte para cima e para baixo. Esse erro é observado quando há um problema de roteamento recursivo.

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

Roteador como participante PMTUD no endpoint do túnel

O roteador tem duas funções PMTUD diferentes quando é o ponto final de um túnel.

- Na primeira função, o roteador é o remetente de um pacote de host. Para o processamento de PMTUD, o roteador precisa verificar o tamanho do bit DF e do pacote de dados original para executar a ação apropriada, quando necessário.

- A segunda função ocorre após o roteador encapsular o pacote IPv4 original do host dentro do pacote de túnel. Nessa fase, o roteador atua mais com um host em relação ao PMTUD e ao pacote IPv4 do túnel.

Quando o roteador atua na primeira função (um roteador que encaminha pacotes IPv4 do host), essa função ocorre antes de o roteador encapsular o pacote IPv4 do host dentro do pacote de túnel.

Se o roteador participar como o remetente de um pacote do host, ele executará estas ações:

- Verifica se o bit DF está definido.

- Verifica qual tamanho de pacote o túnel pode comportar.

- Fragmentar (se o pacote for muito grande e o bit DF não estiver definido), encapsular os fragmentos e enviar ou

- Descarta o pacote (se o pacote for muito grande e o bit DF estiver definido) e envia uma mensagem de ICMP para o remetente.

- Encapsula (se o pacote não for muito grande) e envia.

Genericamente, há uma escolha de encapsulamento e fragmentação (envia dois fragmentos de encapsulamento) ou fragmentação e encapsulamento (envia dois fragmentos encapsulados).

Os dois exemplos que mostram a interação de PMTUD e pacotes que passam pelas redes de exemplo são detalhados nesta seção.

O primeiro exemplo mostra o que acontece com um pacote quando o roteador (na origem do túnel) tem a função de roteador de encaminhamento.

Para processar o PMTUD, o roteador precisa verificar o tamanho do bit DF e do pacote de dados original para executar a ação apropriada.

Este exemplo usa encapsulamento GRE para o túnel. O GRE faz a fragmentação antes do encapsulamento.

Os exemplos posteriores mostram cenários em que a fragmentação é feita após o encapsulamento.

No exemplo 1, o bit DF não está definido (DF = 0) e a MTU de IPv4 do túnel GRE é 1.476 (1.500-24).

Exemplo 1

1. O roteador de encaminhamento (na fonte de túnel) recebe um datagrama de 1.500 bytes com o bit de DF limpo (DF= 0) do host remetente.

Esse datagrama é composto por um cabeçalho de IP de 20 bytes e um payload de TCP de 1480 bytes.

IPv4	TCP de 1.480 bytes + dados

2. Como o pacote é muito grande para o IPv4 de MTU depois da adição do overhead de GRE (24 bytes), o roteador de encaminhamento divide o datagrama em dois fragmentos de 1.476 (20 bytes de cabeçalho IPv4 + 1.456 bytes de payload IPv4) e 44 bytes (20 bytes de cabeçalho IPv4 + 24 bytes de payload IPv4)

Depois que o encapsulamento GRE for adicionado, o pacote não será maior do que o MTU de interface física de saída.

IP0	TCP de 1.456 bytes + dados
IP1	Dados de 24 bytes

3. O roteador de encaminhamento adiciona o encapsulamento GRE, incluindo um cabeçalho GRE de 4 bytes e um cabeçalho IPv4 de 20 bytes, a cada fragmento do datagrama IPv4 original.

Esses dois datagramas IPv4 agora têm um comprimento de 1.500 e 68 bytes e não são considerados fragmentos, mas sim datagramas IP individuais.

IPv4	GRE	IP0	TCP de 1.456 bytes + dados
IPv4	GRE	IP1	Dados de 24 bytes

4. O roteador de destino do túnel remove o encapsulamento GRE de cada fragmento do datagrama original, permanecendo dois fragmentos IPv4 de 1.476 e 24 bytes de comprimento.

Esses fragmentos de datagrama de IPv4 são encaminhados separadamente por este roteador para o host destinatário.

IP0	TCP de 1.456 bytes + dados
IP1	Dados de 24 bytes

5. O host destinatário remonta esses dois fragmentos no datagrama original.

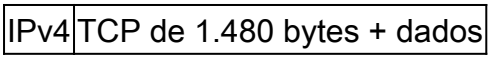
IPv4	TCP de 1.480 bytes + dados

O exemplo 2 descreve a função do roteador de encaminhamento no contexto de uma topologia de rede.

O roteador tem a mesma função do roteador de encaminhamento, mas dessa vez o bit DF está definido (DF = 1).

Exemplo 2

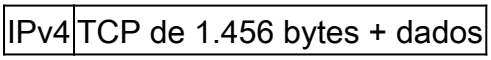
1. O roteador de encaminhamento na fonte de túnel recebe um datagrama de 1.500 bytes com DF = 1 proveniente do host remetente.



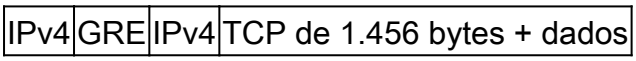
2. Como o bit DF está definido, e o tamanho do datagrama (1.500 bytes) é maior do que o MTU IPv4 do túnel GRE (1476), o roteador descarta o datagrama e envia uma mensagem "ICMP fragmentation needed but DF bit set" para a origem do datagrama.

A mensagem ICMP alerta o remetente de que o MTU é 1.476.

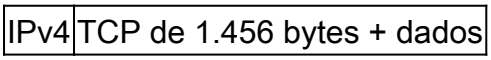
3. O host remetente recebe a mensagem de ICMP e, ao reenviar os dados originais, ele usa uma datagrama IPv4 de 1.476 bytes.



4. Esse comprimento de datagrama IPv4 (1.476 bytes) agora é igual em valor ao MTU IPv4 do túnel GRE, de modo que o roteador adiciona o encapsulamento GRE ao datagrama IPv4.



5. O roteador destinatário (no destino do túnel) remove o encapsulamento GRE do datagrama IPv4 e o envia para o host destinatário.



É isso que acontece quando o roteador tem uma segunda função como host remetente em relação ao PMTUD e ao pacote IPv4 do túnel.

Essa função ocorre após o roteador encapsular o pacote IPv4 original do host dentro do pacote de túnel.

Observação: por padrão, um roteador não faz PMTUD nos pacotes de túnel GRE gerados por ele. O comando**tunnel path-mtu-discovery** pode ser usado para ativar o PMTUD para pacotes de túnel GRE-IPv4.

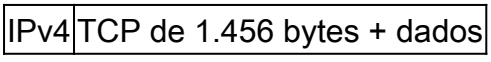
O exemplo 3 mostra o que acontece quando o host envia datagramas IPv4 suficientemente pequenos para caber no IPv4 do MTU da interface de túnel GRE.

O bit DF, nesse caso, pode ser configurado ou limpo (1 ou 0).

A interface de túnel GRE não tem o **tunnel path-mtu-discovery** comando configurado, portanto o roteador não morre PMTUD no pacote GRE-IPv4.

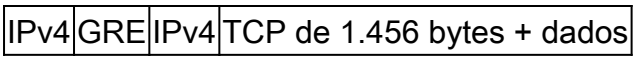
Exemplo 3

1. O roteador de encaminhamento na fonte de túnel recebe um datagrama de 1.476 bytes do host remetente.



2. Esse roteador encapsula o datagrama de IPv4 de 1476 bytes dentro do GRE para obter um datagrama de IPv4 do GRE de 1.500 bytes.

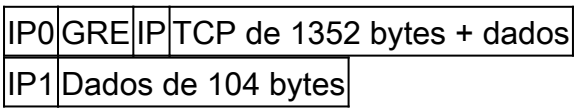
O bit DF no cabeçalho do IPv4 do GRE é removido (DF = 0). Esse roteador encaminha, em seguida, esse pacote ao destino do túnel.



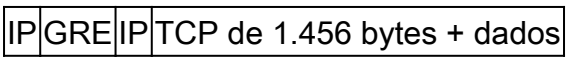
3. Pressuponha que há um roteador entre a origem e o destino do túnel com um MTU de link de 1.400.

Este roteador fragmenta o pacote de túnel, pois o bit DF foi removido (DF = 0).

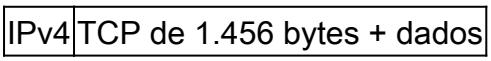
Lembre-se que esse exemplo fragmenta o IPv4 mais externo, então os cabeçalhos de GRE, IPv4 interno e TCP só aparecem no primeiro fragmento.



4. O roteador de destino do túnel deve reagrupar o pacote de túnel GRE.



5. Depois que o pacote de túnel GRE for remontado, o roteador removerá o cabeçalho IPv4 de GRE e encaminhará normalmente o datagrama IPv4 original.

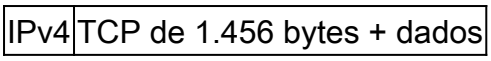


O exemplo 4 mostra o que acontece quando o roteador funciona como host remetente em relação ao PMTUD e ao pacote IPv4 do túnel.

Dessa vez, o bit DF é definido (DF = 1) no cabeçalho IPv4 original e o **tunnel path-mtu-discovery** comando foi configurado para que o bit DF seja copiado do cabeçalho IPv4 interno para o cabeçalho externo (GRE + IPv4).

Exemplo 4

1. O roteador de encaminhamento na fonte de túnel recebe um datagrama de 1.476 bytes com DF = 1 proveniente do host remetente.



2. Esse roteador encapsula o datagrama de IPv4 de 1476 bytes dentro do GRE para obter um datagrama de IPv4 do GRE de 1.500 bytes.

Este cabeçalho IPv4 de GRE tem o DF bit definido (DF = 1), pois o datagrama IPv4 original tinha o bit DF definido.

Esse roteador encaminha, em seguida, esse pacote ao destino do túnel.

3. Novamente, pressuponha que há um roteador entre a origem e o destino do túnel com um MTU de link de 1.400.

Este roteador não fragmenta o pacote de túnel, pois o bit DF foi definido (DF = 1).

Esse roteador deve descartar o pacote e enviar uma mensagem de erro do ICMP para o roteador de origem do túnel, pois esse é o endereço IPv4 de origem no pacote.

IPv4	MTU de ICMP de 1.400

4. O roteador de encaminhamento na origem do túnel recebe esta mensagem de erro do "ICMP" e diminui o MTU de IPv4 do túnel GRE para 1.376 (1.400 - 24).

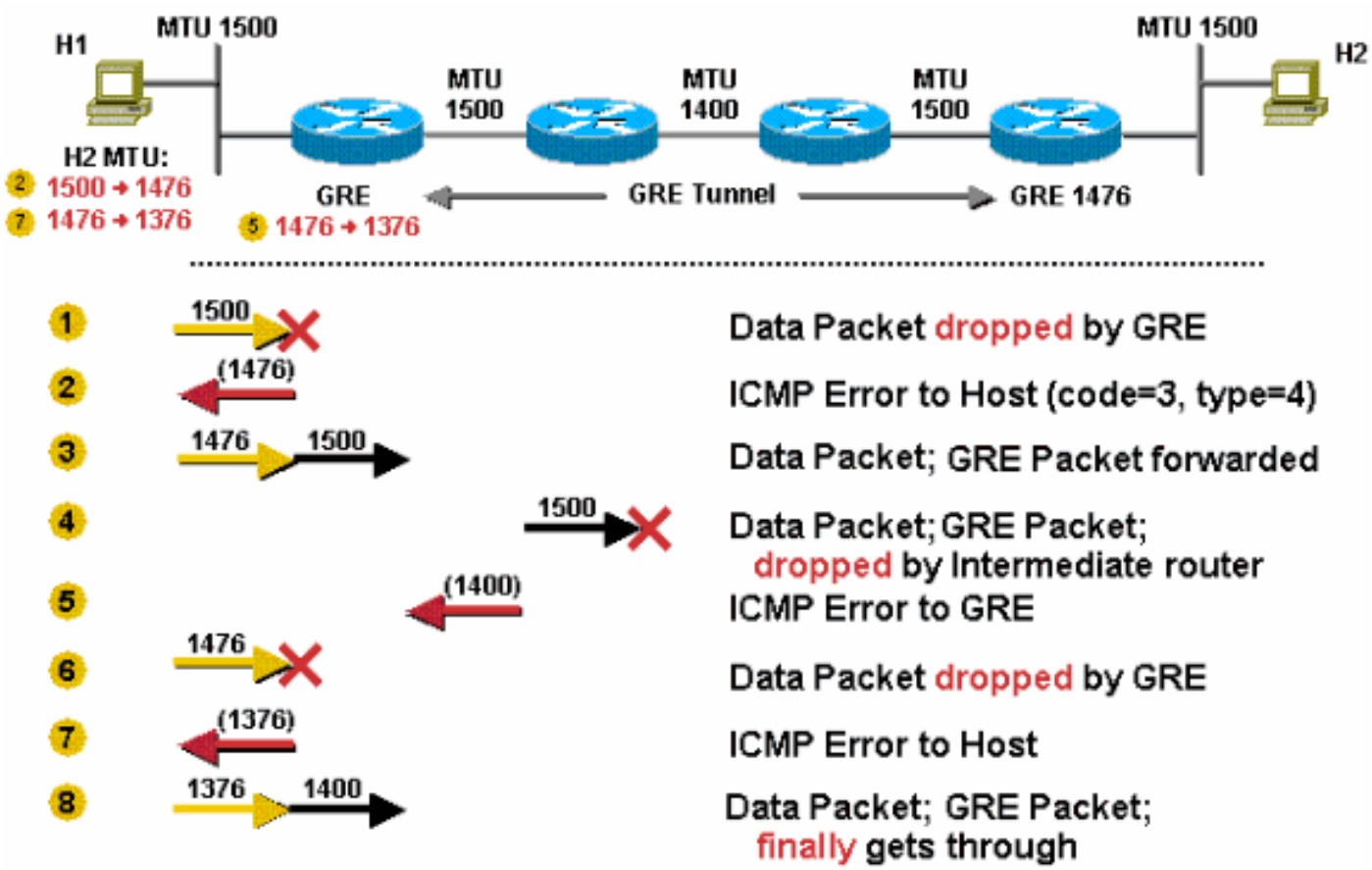
Na próxima vez que o host de envio retransmite os dados em um pacote de IPv4 de 1.476 bytes, esse pacote pode ser muito grande e o roteador envia uma mensagem de erro do "ICMP" para o remetente com um valor de MTU de 1.376.

Quando o host remetente retransmite os dados, ele envia um pacote IPv4 de 1.376 bytes e o pacote atravessa o túnel GRE até o host destinatário.

Exemplo 5

Este exemplo ilustra a fragmentação do GRE. Fragmente antes do encapsulamento para o GRE e, em seguida, faça o PMTUD para o pacote de dados; além disso, o bit DF não é copiado quando o pacote IPv4 é encapsulado pelo GRE.

O bit DF não foi definido. O MTU de IPv4 da interface de túnel GRE é, por padrão, 24 bytes menor do que o MTU de IPv4 de interface física, portanto o MTU de IPv4 da interface GRE é 1.476, como mostrado na imagem.



- O remetente envia um pacote de 1.500 bytes (cabeçalho IPv4 de 20 bytes + 1.480 bytes de payload de TCP).

- Como o MTU do túnel GRE é 1.476, o pacote de 1500 bytes é dividido em dois fragmentos IPv4 de 1.476 e 44 bytes, cada um prevendo os 24 bytes adicionais do cabeçalho GRE.

- Os 24 bytes do cabeçalho GRE são adicionados a cada fragmento de IPv4. Agora os fragmentos têm 1.500 (1.476 + 24) e 68 (44 + 24) bytes cada.

- Os pacotes IP + GRE que contêm os dois fragmentos IPv4 são encaminhados para o roteador de túnel GRE correspondente.

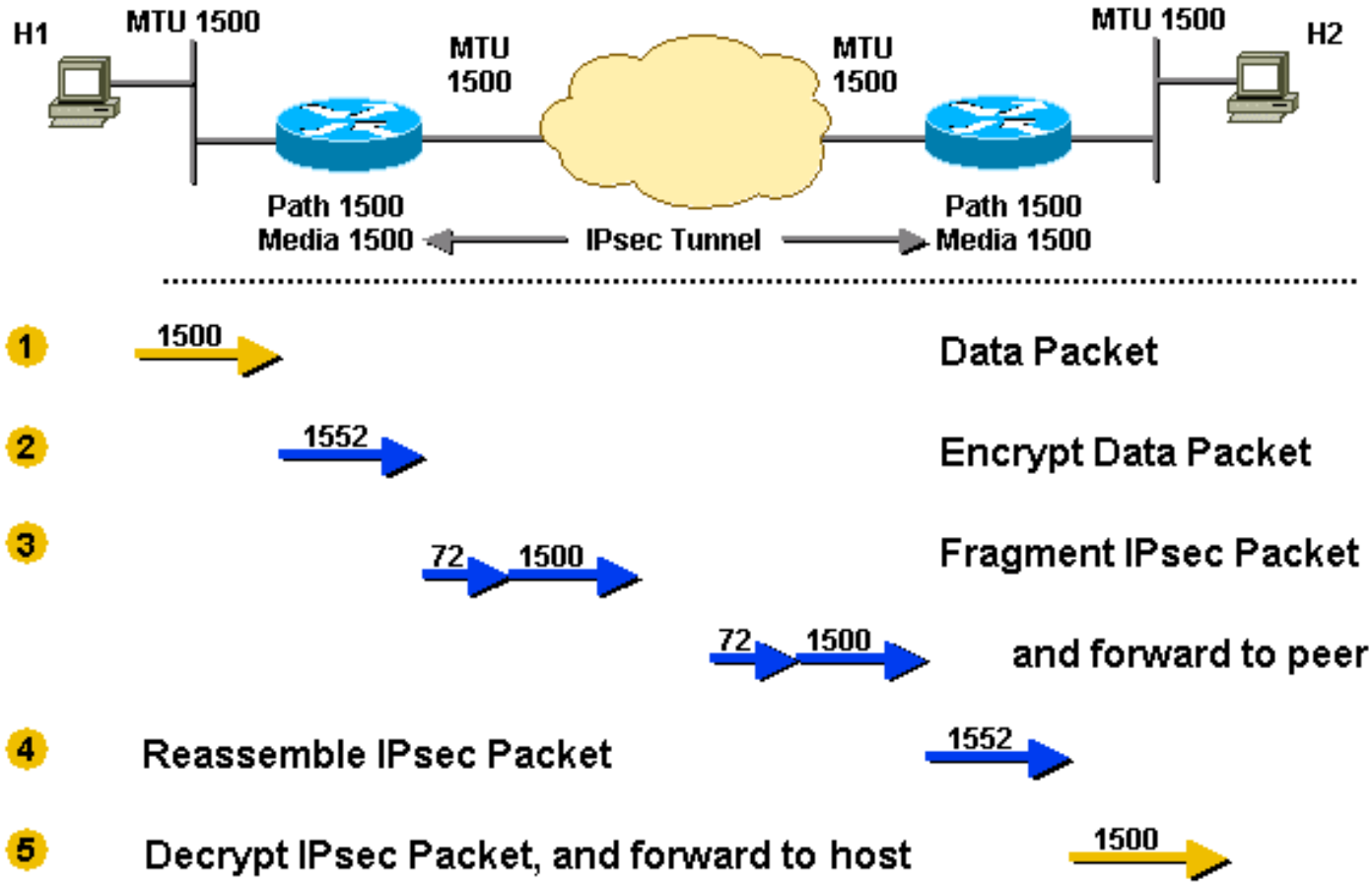
- O roteador do par do túnel GRE remove os cabeçalhos GRE dos dois pacotes.
- Esse roteador encaminha os dois pacotes para o host de destino.
- O host de destino reagrupa os fragmentos IPv4 de volta ao datagrama IP original.

Exemplo 6

Esse exemplo é semelhante ao exemplo 5, mas desta vez o bit DF foi definido. O roteador é configurado para fazer PMTUD em pacotes de túnel GRE + IPv4 com o **tunnel path-mtu-discovery**comando e o bit DF é copiado do cabeçalho IPv4 original para o cabeçalho IPv4 do GRE.

Se o roteador recebe um erro de ICMP para o pacote IPv4 + GRE, ele reduz o MTU do IP na interface de túnel GRE.

O MTU do IPv4 do túnel GRE é definido como 24 bytes, menos do que o MTU de interface física por padrão, portanto, esse MTU de IPv4 de GRE é 1.476. Há um link de MTU de 1.400 no caminho do túnel GRE, conforme mostrado na imagem.



- O roteador recebe um pacote de 1.500 bytes (cabeçalho IPv4 de 20 bytes + payload TCP de 1.480), e ele descarta o pacote. O roteador descarta o pacote porque é maior do que o IPv4 de MTU (1.476) na interface de túnel GRE.
- O roteador envia um erro ICMP ao remetente informando que o próximo MTU de nó é 1476. O host registra esta informação, geralmente como uma rota de host para o destino, em sua tabela de roteamento.
- O host de envio usa um tamanho de pacote de 1.476 bytes quando reenvia os dados. O roteador GRE acrescenta 24 bytes de encapsulamento de GRE e envia um pacote de 1500 bytes.

- O pacote de 1500 bytes não pode atravessar o enlace de 1400 bytes; portanto, será descartado pelo roteador intermediário.

- O roteador intermediário envia um ICMP (tipo = 3, código = 4) para o roteador GRE com um MTU de próximo salto de 1.400. O roteador GRE reduz isso a 1.376 (1.400-24) e define um valor de MTU de IPv4 interno na interface GRE. Essa alteração só pode ser vista quando você usa o **debug tunnel** comando; ela não pode ser vista na saída do **show ip interface tunnel<#>** comando.

- Da próxima vez que o host enviar novamente o pacote de 1.476 bytes, o roteador GRE descartará o pacote, já que é maior do que o MTU de IPv4 (1.376) atual na interface de túnel GRE.

- O roteador GRE envia outro ICMP (tipo = 3, código = 4) ao remetente com um MTU de próximo salto de 1.376 e o host atualiza as próprias informações atuais com o novo valor.

- O host reenvia novamente os dados, mas agora em um pacote menor de 1.376 bytes, o GRE adiciona 24 bytes de encapsulamento e encaminha. Desta vez, o pacote chega ao par do túnel GRE, onde é desencapsulado e enviado ao host de destino.

Observação: **se o tunnel path-mtu-discovery** comando não foi configurado no roteador de encaminhamento nesse cenário, e o bit DF foi definido nos pacotes encaminhados pelo túnel GRE, o Host 1 ainda conseguirá enviar pacotes TCP/IPv4 para o Host 2, mas eles serão fragmentados no meio no link MTU 1400. Além disso, o par do túnel GRE tem que remontá-los, antes de serem desencapsulados, e enviá-los.

GRE + IPsec (Modo de túnel)

O protocolo de segurança IPv4 (IPv4sec) é um método padronizado que fornece privacidade, integridade e autenticidade às informações transferidas pelas redes IPv4.

O IPv4sec fornece a criptografia de camada de rede IPv4. O IPv4sec alonga o pacote IPv4 ao adicionar pelo menos um cabeçalho IPv4 (modo de túnel).

Os cabeçalhos adicionados variam de comprimento, dependendo do modo de configuração do IPv4sec, mas não ultrapassam cerca de 58 bytes (Encapsulating Security Payload - ESP e ESP authentication - ESPauth) por pacote.

O IPv4sec tem dois modos, o modo de túnel e o modo de transporte.

- Modo de túnel é o modo padrão. No modo de túnel, todo pacote IPv4 original é protegido (criptografado, autenticado ou ambos) e encapsulado pelos cabeçalhos e rodapés IPv4sec. Em seguida, um novo cabeçalho IPv4 é anexado ao pacote, especificando os endpoints de IPv4sec (pares) como a origem e o destino. O modo de túnel pode ser usado com qualquer tráfego IPv4 unicast e deve ser usado se o IPv4sec protege o tráfego contra hosts escondidos nos pares IPv4sec. Por exemplo, o modo de túnel é usado em Redes virtuais privadas (VPNs), onde hosts em uma rede protegida enviam pacotes para hosts em uma rede diferente por meio de um par de pares IPv4sec. Com VPNs, o "túnel" IPv4sec protege o tráfego de IPv4 entre os hosts ao criptografar esse tráfego entre os roteadores de pares IPv4sec.

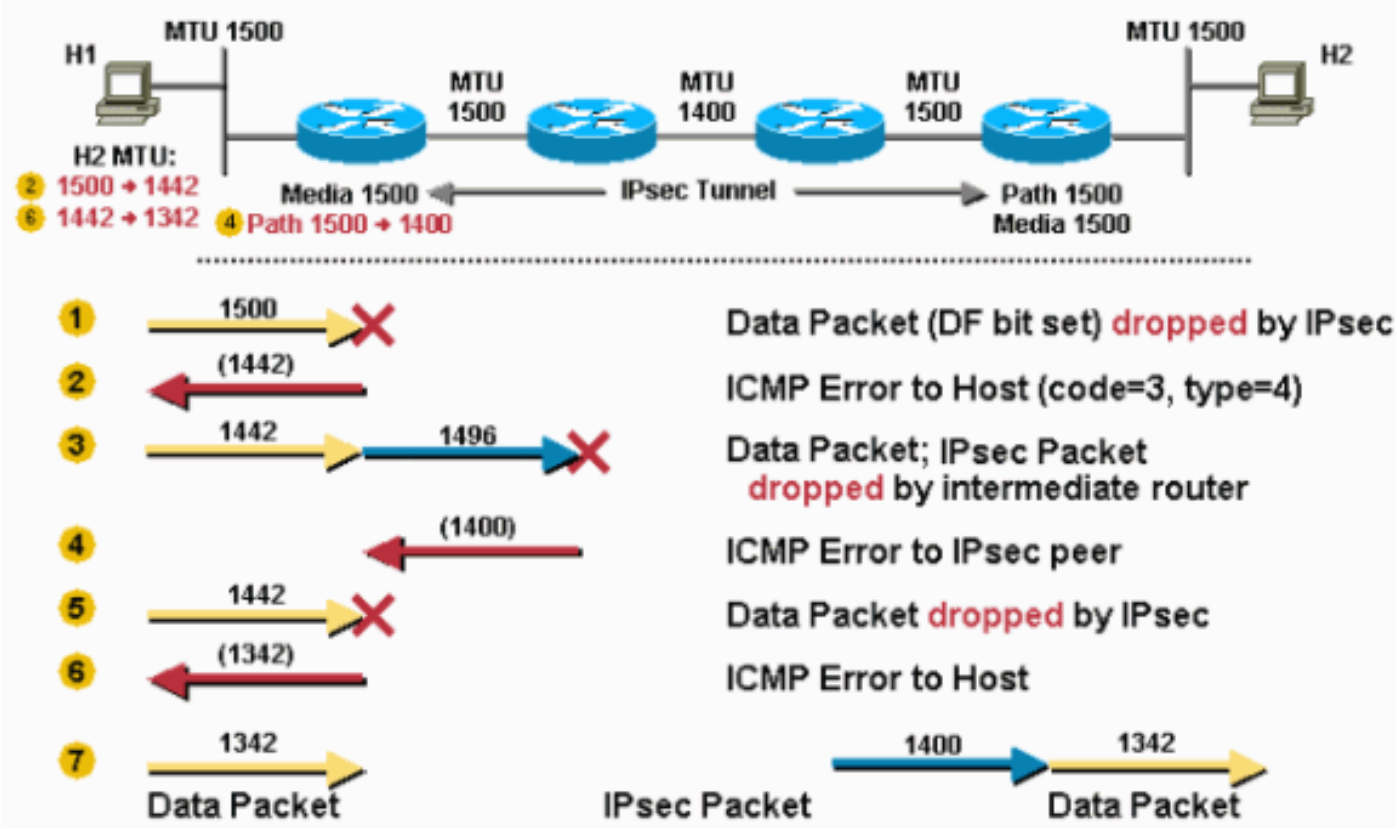
- Com o modo de transporte (configurado com o subcomando, **mode transport**, na definição de transformação), somente o payload do pacote IPv4 original é protegido (criptografado, autenticado ou ambos). O payload é encapsulado pelos cabeçalhos e trailers de IPv4sec. Os cabeçalhos IPv4 originais permanecem intactos, exceto o campo de protocolo IPv4 que é alterado para ESP (50), e o valor do protocolo original é salvo no trailer IPv4sec para ser restaurado quando o pacote for descriptografado. O modo de transporte é usado apenas quando o tráfego IPv4 a ser protegidos está entre os pares de IPv4sec, a origem e os endereços IPv4 de destino no pacote são os mesmos que os endereços de mesmo nível de IPv4sec. Normalmente o modo de transporte IPv4sec é usado somente quando um outro protocolo de encapsulamento (como GRE) for usado para encapsular primeiro o pacote de dados IPv4, em seguida, o IPv4sec é usado para proteger os pacotes de túnel GRE.

O IPv4sec sempre faz PMTUD para pacotes de dados e para seus próprios pacotes. Existem comandos de configuração IPv4sec para modificar o processamento de PMTUD para o pacote IPv4 de IPv4sec, o IPv4sec pode limpar, definir ou copiar o bit DF do cabeçalho IPv4 do pacote de dados para o cabeçalho IPv4 de IPv4sec. Esse recurso é denominado "Funcionalidade de Anulação de Bit DF".

Note: evite a fragmentação após o encapsulamento ao fazer a criptografia de hardware com o IPv4sec. A criptografia de hardware fornece a você uma produtividade de cerca de 50 Mbs, dependendo do hardware. Porém, se o pacote IPv4sec estiver fragmentado, você perderá de 50 a 90% da produtividade. Essa perda ocorre porque os pacotes IPv4sec fragmentados são comutados por processo para remontagem e, em seguida, enviados para o mecanismo de criptografia de hardware para descriptografia. Essa perda de produtividade pode prejudicar a produtividade de criptografia de hardware até o nível de desempenho da criptografia de software (2-10 Mbs).

Exemplo 7

Este cenário retrata a fragmentação de IPv4sec em ação. Nesse cenário, o MTU em todo o caminho é 1.500. Nesse cenário, o bit DF não está definido.



- O roteador recebe um pacote de 1.500 bytes (cabeçalho IPv4 de 20 bytes + payload TCP de 1.480) destinado para o Host 2.

- O pacote de 1.500 bytes é criptografado por IPv4sec e 52 bytes de sobrecarga são adicionados (cabeçalho IPv4sec, trailer e cabeçalho adicional do IPv4). Agora, o IPv4sec precisa enviar um pacote de 1.552 bytes. Como o MTU de saída é 1.500, esse pacote precisa ser fragmentado.

- Dois fragmentos são criados fora do pacote IPv4sec. Durante a fragmentação, outro cabeçalho IPv4 de 20 bytes é adicionado ao segundo fragmento, que resulta em um fragmento de 1.500 bytes e um fragmento IPv4 de 72 bytes.

- O roteador de par do túnel IPv4sec recebe os fragmentos, remove o cabeçalho IPv4 adicional e agrupa os fragmentos IPv4 de volta ao pacote original do IPv4sec. Em seguida, o IPv4sec descriptografa esse pacote.

- O roteador encaminha o pacote de dados original de 1500 bytes para o Host 2.

Exemplo 8

Esse exemplo é semelhante ao exemplo 6, porém, neste caso, o bit DF está definido no pacote de dados original e há um link no caminho entre os pares de túnel IPv4sec com MTU menor do que os outros links.

Esse exemplo demonstra como o roteador par de IPv4sec executa as duas funções de PMTUD, conforme descrito na seção O roteador como um participante PMTUD no endpoint de um túnel.

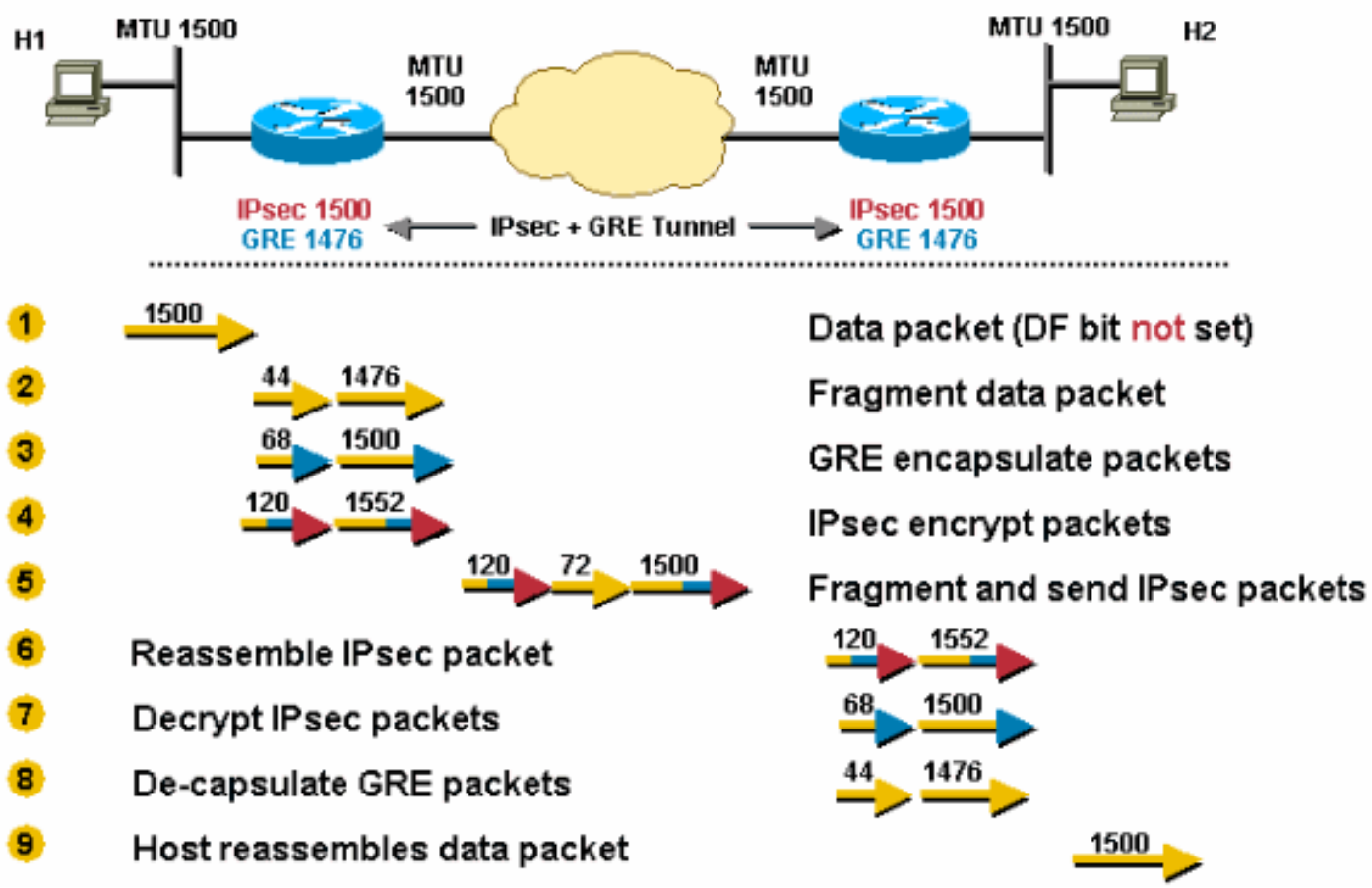
O PMTU de IPv4sec é alterado para um valor mais baixo, como resultado da necessidade de fragmentação.

O bit DF é copiado do cabeçalho IPv4 interno para o cabeçalho IPv4 externo, quando o IPv4sec criptografa um pacote.

Os valores de MTU e PMTU de mídia são armazenados em Security Association (SA) do IPv4sec.

A mídia de MTU baseia-se no MTU da interface do roteador de saída e o PMTU tem como base o MTU mínimo visto no caminho entre os pares de IPv4sec.

O IPv4sec encapsula/criptografa o pacote antes de tentar fragmentá-lo, conforme mostrado na imagem.



- O roteador recebe um pacote de 1.500 bytes e o descarta porque o overhead de IPv4sec, quando adicionado, deixa o pacote maior que o PMTU (1.500).

- O roteador envia uma mensagem ICMP ao Host 1 informando-o de que o MTU do próximo salto é 1442 (1500 - 58 = 1442). Esses 58 bytes são a sobrecarga de IPv4sec máxima ao usar ESP e ESPauth de IPv4sec. A sobrecarga real do IPv4sec pode ser até 7 bytes menor que esse valor. O Host 1 registra esta informação, geralmente como uma rota de host para o destino (Host 2), em sua tabela de roteamento.

- O Host 1 reduz o PMTU do Host 2 para 1.442, então o Host 1 envia pacotes menores (1.442 bytes) quando retransmitir os dados para o Host 2. O roteador recebe o pacote 1.442 bytes e o IPv4sec adiciona 52 bytes de sobrecarga de criptografia, então o pacote IPv4sec resultante é de 1.496 bytes. Como esse pacote tem o DF bit definido no cabeçalho, ele é descartado pelo roteador do meio com o link de MTU de 1.400 bytes.

- O roteador do meio que descartou o pacote envia uma mensagem ICMP para o remetente do pacote IPv4sec (o primeiro roteador) dizendo que o MTU do próximo salto é de 1.400 bytes. Esse valor é registrado no PMTU de SA do IPv4sec.

- Na próxima vez que Host 1 retransmitir o pacote de 1.442 bytes (ele não recebeu uma confirmação disso), o IPv4sec descartará o pacote. O roteador descarta o pacote porque a sobrecarga de IPv4sec, quando adicionado ao pacote, vai torná-lo maior do que o PMTU (1.400).

- O roteador envia uma mensagem ICMP para o Host 1 dizendo que o MTU do próximo salto agora é de 1.342. (1.400 - 58 = 1.342). O Host 1 grava essa informação novamente.

- Quando o Host 1 retransmite novamente os dados, ele usa o pacote de tamanho menor (1.342). Esse pacote não exige fragmentação e atravessa o túnel IPv4sec até o Host 2.

GRE e IPv4sec juntos

As interações mais complexas para fragmentação e PMTUD ocorrem quando o IPv4sec é usado para criptografar túneis GRE.

O IPv4sec e o GRE são combinados dessa maneira porque o IPv4sec não comporta pacotes de multicast de IPv4, o que significa que você não pode executar um protocolo de roteamento dinâmico pela rede IPv4sec VPN.

Os túneis GRE oferecem suporte ao multicast, portanto, um túnel GRE pode ser usado para encapsular primeiro o pacote de multicast do protocolo de roteamento dinâmico em um pacote de unicast de IPv4 de GRE, que pode então ser criptografado pelo IPv4sec.

Ao fazer isso, o IPv4sec geralmente é implementado no modo de transporte no topo do GRE, porque os pares de IPv4sec e os endpoints de túnel de GRE (os roteadores) são os mesmos, e o modo de transporte salva 20 bytes de sobrecarga de IPv4sec.

Um caso interessante é quando um pacote IPv4 tiver sido dividido em dois fragmentos e encapsulado pelo GRE.

Nesse caso, o IPv4sec vê dois pacotes de GRE + IPv4 independentes. Frequentemente, em uma configuração padrão, um desses pacotes é grande o suficiente para precisar ser fragmentado depois que for criptografado.

O par de IPv4sec tem que remontar este pacote antes de descriptografar. Essa “fragmentação dupla" (uma vez antes de GRE e novamente depois de IPv4sec) no roteador remetente aumenta a latência e reduz a taxa de transferência.

A remontagem ocorre em switch de processo, então há um acesso à CPU no roteador destinatário sempre que isso acontece.

Essa situação pode ser evitada ao definir o "ip mtu" da interface do túnel GRE baixo o suficiente para levar em conta a sobrecarga do GRE e IPv4sec (por padrão, a interface do túnel GRE "ip mtu" é definido como os bytes de sobrecarga de MTU - GRE da interface real de saída).

Esta tabela lista os valores de MTU sugeridos para cada combinação de túnel/modo, pressupondo que a interface física de saída tenha um MTU de 1.500.

Combinação de Túneis	MTU específico necessário	MTU recomendado
GRE + IPsec (Modo de transporte)	1440 bytes	1400 bytes
GRE + IPsec (Modo de túnel)	1420 bytes	1400 bytes

Note: o valor de MTU de 1.400 é recomendado porque abrange as combinações de modo de GRE + IPv4sec mais comuns. Além disso, não há nenhuma desvantagem discernível em permitir um overhead de 20 ou 40 bytes adicionais. É mais fácil de lembrar e definir um valor e esse valor abrange quase todos os cenários.

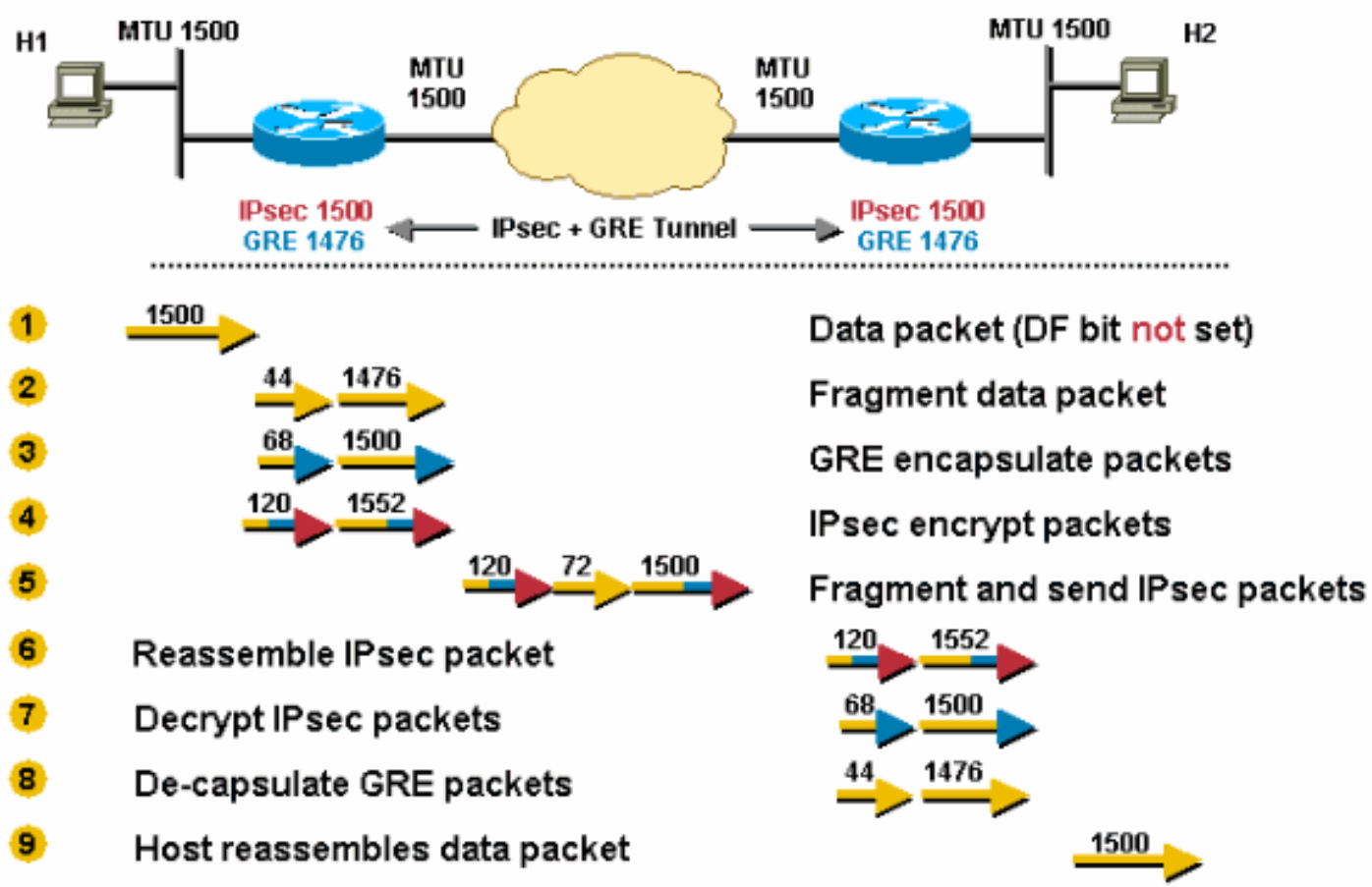
Exemplo 9

O IPv4sec é implantado por cima do GRE. O MTU físico de saída é 1.500, o IPv4sec de PMTU é 1.500 e o MTU de IPv4 de GRE é 1.476 (1.500-24 = 1.476).

Os pacotes TCP/IPv4 são, portanto, fragmentados duas vezes, uma vez antes do GRE e uma vez depois do IPv4sec.

O pacote é fragmentado antes do encapsulamento de GRE e um desses pacotes GRE é fragmentado novamente após a criptografia de IPv4sec.

Configurar o "ip mtu 1440" (modo de transporte de IPv4sec) ou "mtu ip 1420" (modo de túnel de IPv4sec) no túnel GRE eliminaria a possibilidade de fragmentação dupla nesse cenário.



- O roteador recebe um datagrama de 1.500 bytes.

- Antes do encapsulamento, o GRE fragmenta o pacote de 1.500 bytes em dois pedaços, 1.476 (1.500-24 = 1.476) e 44 (24 dados + 20 cabeçalhos IPv4) bytes.

- O GRE encapsula os fragmentos IPv4, acrescentando 24 bytes a cada pacote. Isso resulta em dois pacotes GRE + IPv4sec de 1.500 (1.476 + 24 = 1.500) e 68 (44 + 24) bytes cada.

- O IPv4sec criptografa os dois pacotes, que adicionam 52 bytes (modo de túnel de IPv4sec) de overhead de encapsulamento para cada um, para fornecer um pacote de 1.552 bytes e um pacote de 120 bytes.

- O pacote IPv4sec de 1.552 bytes é fragmentado pelo roteador porque é maior do que o MTU de saída (1.500). O pacote de 1.552 bytes é dividido em partes, um pacote de 1.500 bytes e um pacote de 72 bytes (52 bytes "de payload" mais um cabeçalho de IPv4 de 20 bytes adicional para o segundo fragmento). Os três pacotes de 1.500 bytes, 72 bytes e 120 bytes são encaminhados para o par IPv4sec + GRE.

- O roteador destinatário remonta a dois fragmentos de IPv4sec (1.500 bytes e 72 bytes) para compor o pacote de IPv4sec de 1.552 bytes + pacote GRE original. Nada precisa ser feito para o pacote IPv4sec + GRE de 120 bytes.

- O IPv4sec descriptografa os pacotes IPv4sec + GRE de 1.552 bytes e 120 bytes para obter pacotes GRE de 68 bytes e de 1.500 bytes.

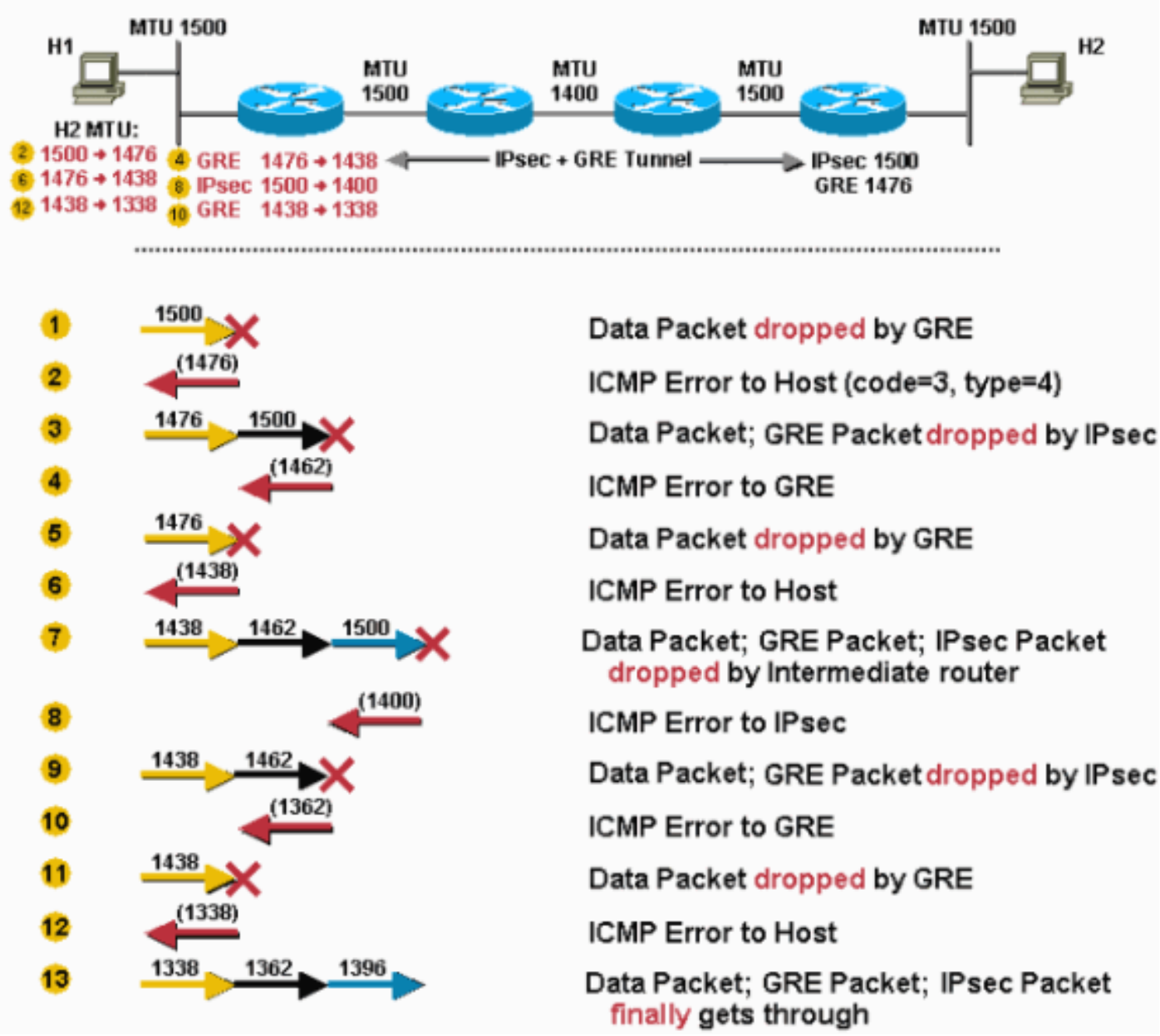
- O GRE desencapsula os pacotes GRE de 1.500 bytes e de 68 bytes para obter fragmentos de pacotes IPv4 de 1.476 bytes e 44 bytes. Esses fragmentos de pacotes IPv4 são encaminhados para o host de destino.

- O Host 2 remonta esses fragmentos IPv4 para obter o datagrama IPv4 original de 1.500 bytes.

O Cenário 10 é semelhante ao Cenário 8, exceto pela existência de um link MTU mais baixo no caminho de túnel. Este é um pior cenário para o primeiro pacote enviado do Host 1 para o Host 2. Após a última etapa nesse cenário, o Host 1 define o PMTU correto para o Host 2 e tudo funciona para as conexões TCP entre os Hosts 1 e 2. Os fluxos TCP entre o Host 1 e outros hosts (acessível por meio do túnel IPv4sec + GRE) só precisa atravessar as três últimas etapas do Cenário 10.

Neste cenário, o **tunnel path-mtu-discovery** comando é configurado no túnel GRE e o bit DF é definido em pacotes TCP/IPv4 que se originam do Host 1.

Exemplo 10



- O roteador recebe um pacote de 1.500 bytes. Este pacote é descartado pelo GRE porque o GRE não pode fragmentá-lo ou encaminhá-lo, pois o bit DF está definido, e o tamanho do pacote excede a interface de saída "ip mtu" depois de adicionar o overhead do GRE (24 bytes).

- O roteador envia uma mensagem ICMP ao Host 1 para informar que o MTU do próximo salto agora é de 1.476 (1.500 - 24 = 1.476)

- O Host 1 altera o PMTU do Host 2 para 1.476 e envia o menor tamanho quando retransmite o pacote. O GRE o encapsula e entrega o pacote de 1.500 bytes para o IPv4sec. O IPv4sec descarta o pacote porque o GRE copiou o bit DF (definido) do cabeçalho IPv4 interno e, com o overhead de IPv4sec (máximo de 38 bytes), o pacote é muito grande para encaminhamento fora da interface física.

- O IPv4sec envia uma mensagem ICMP para GRE que indica que o MTU do próximo salto é de 1.462 bytes (já que um máximo de 38 bytes é adicionado para criptografia e overhead de IPv4). O GRE registra o valor 1.438 (1.462-24) como o "ip mtu" da interface de túnel.

- **Observação**: essa alteração no valor é armazenada internamente e não pode ser vista na saída do **show ip interface tunnel**<#>comando. Você verá essa alteração apenas se acionar o **debug tunnel** comando.

- Da próxima vez que o Host 1 retransmitir o pacote de 1.476 bytes, ele será descartado pelo GRE.
- O roteador envia uma mensagem ICMP para o Host 1, que indica que 1.438 é o MTU do próximo salto.
- O Host 1 reduz o PMTU para o Host 2 e retransmite um pacote de 1.438 bytes. Desta vez, o GRE aceita o pacote, o encapsula e entrega ao IPv4sec para criptografia.
- O pacote de IPv4sec é encaminhado para o roteador intermediário e descartado porque tem uma interface de saída MTU de 1.400.
- O roteador intermediário envia uma mensagem ICMP para o IPv4sec dizendo que o MTU do próximo salto é de 1.400. Esse valor é gravado pelo IPv4sec no valor de PMTU do SA de IPv4sec associado.
- Quando o Host 1 retransmite o pacote de 1.438 bytes, o GRE o encapsula e entrega para o IPv4sec. O IPv4sec descarta o pacote porque mudou seu próprio PMTU para 1.400.
- O IPv4sec envia um erro de ICMP ao GRE, indicando que o MTU do próximo salto é de 1.362, e o GRE registra o valor 1.338 internamente.
- Quando o Host 1 retransmite o pacote original (porque não recebeu uma confirmação de recebimento), ele é descartado pelo GRE.
- O roteador envia uma mensagem ICMP ao Host 1, indicando que o MTU do próximo salto é de 1.338 (1.362 - 24 bytes). O Host 1 reduz para 1.338 o PMTU para o Host 2.
- O Host 1 retransmite um pacote de 1.338 bytes e dessa vez pode finalmente chegar até o fim para o Host 2.

Outras recomendações

Configurar o **tunnel path-mtu-discovery** comando em uma interface de túnel pode ajudar na interação de GRE e IPv4sec quando eles estão configurados no mesmo roteador.

Sem o **tunnel path-mtu-discovery** comando configurado, o bit DF sempre seria apagado no cabeçalho IPv4 do GRE.

Essa configuração permite que o pacote IPv4 de GRE seja fragmentado, mesmo que o cabeçalho IPv4 dos dados encapsulados tenha o bit DF definido, o que normalmente não permitiria a fragmentação do pacote.

Se o **tunnel path-mtu-discovery** comando estiver configurado na interface de túnel GRE:

- O GRE copia o bit DF do cabeçalho IPv4 de dados para o cabeçalho IPv4 do GRE.
- Se o bit DF estiver definido no cabeçalho IPv4 do GRE e o pacote for "muito grande" após a criptografia IPv4sec para o MTU de IPv4 na interface de saída física, então o IPv4sec descartará o pacote e notificará ao túnel GRE que reduza seu tamanho de MTU de IPv4.
- O IPv4sec faz o PMTUD para seus próprios pacotes e se o PMTU do IPv4sec muda (caso seja reduzido), então o IPv4sec não notifica imediatamente o GRE, mas quando outro pacote maior passa, ocorre o processo da etapa 2.
- O MTU de IPv4 do GRE agora é menor, então descarta quaisquer pacotes de dados IPv4 com o bit DF definido que for muito grande e envia uma mensagem ICMP para o host remetente.

O **tunnel path-mtu-discovery** comando ajuda a interface GRE a definir seu MTU IPv4 dinamicamente, em vez de estaticamente com o **ip mtu** comando. Na verdade, recomenda-se que os dois comandos sejam usados.

O **ip mtu** comando é usado para fornecer espaço para a sobrecarga de GRE e IPv4sec relativa ao MTU IPv4 da interface física de saída local.

O **tunnel path-mtu-discovery** comando permite que a MTU IPv4 do túnel GRE seja reduzida ainda mais se houver um link de MTU IPv4 mais baixo no caminho entre os pares IPv4sec.

Aqui estão algumas ações que podem ser feitas caso você esteja com problemas de PMTUD em uma rede com túneis GRE + IPv4sec configurados.

Esta lista começa com a solução mais desejável.

- Corrija o problema do PMTUD que não está funcionando, geralmente causado por um roteador ou firewall que bloqueia o ICMP.
- Use o **ip tcp adjust-mss** comando nas interfaces de túnel para que o roteador reduza o valor TCP MSS no pacote TCP SYN. Isso ajuda os dois hosts finais (o remetente e o destinatário TCP) a usar pacotes pequenos o suficiente para que o PMTUD não seja necessário.
- Use o roteamento de política na interface de ingresso do roteador e configure um mapa de rota para liberar o bit DF no cabeçalho IPv4 de dados antes de chegar à interface de túnel GRE. Esse aumento permite que o pacote IPv4 de dados seja fragmentado antes do encapsulamento de GRE.
- Aumente o "ip mtu" da interface de túnel GRE para que seja igual ao MTU da interface de saída. Esse aumento permite que o pacote IPv4 de dados seja encapsulado pelo GRE sem fragmentá-lo primeiro. O pacote GRE é criptografado pelo IPv4sec e, em seguida, fragmentado para sair da interface física de saída. Nesse caso, você não configuraria o **tunnel path-mtu-discovery** comando na interface de túnel GRE. Isso pode reduzir significativamente a taxa de transferência porque remontagem do pacote IPv4 no peer IPv4sec é feita no modo de switching de processos.

Informações Relacionadas

- **Página de Suporte do IP Routing**
- **Página de suporte do IPSec (protocolo de segurança IP)**
- **Descoberta de MTU de caminho RFC 1191**
- **Opções de descoberta de MTU RFC 1063 IP**
- **Protocolo de Internet RFC 791**
- **Protocolo de controle de transmissão RFC 793**
- **RFC 879 - O tamanho máximo do segmento de TCP e tópicos relacionados**
- **RFC 1701 Generic Routing Encapsulation (GRE)**
- **Esquema 1241 A de RFC para um protocolo de encapsulamento de Internet**

- **RFC 2003 IP Encapsulation within IP**

- **Suporte Técnico e Documentação - Cisco Systems**

Sobre esta tradução
A Cisco traduziu este documento com a ajuda de tecnologias de tradução automática e humana para oferecer conteúdo de suporte aos seus usuários no seu próprio idioma, independentemente da localização.

Observe que mesmo a melhor tradução automática não será tão precisa quanto as realizadas por um tradutor profissional.

A Cisco Systems, Inc. não se responsabiliza pela precisão destas traduções e recomenda que o documento original em inglês (link fornecido) seja sempre consultado.