Résoudre les problèmes de fragmentation IP, MTU, MSS et PMTUD avec GRE et IPSEC

Contenu

Introduction

Le document décrit comment la détection de fragmentation IP et de Maximum Transmission Unit de chemin (PMTUD) fonctionnent et discutent également quelques scénarios qui comportent le comportement de PMTUD une fois combinés avec différentes combinaisons des tunnels IP. L'utilisation répandue en cours des tunnels IP en Internet a apporté les problèmes qui impliquent la fragmentation IP et le PMTUD au premier rang.

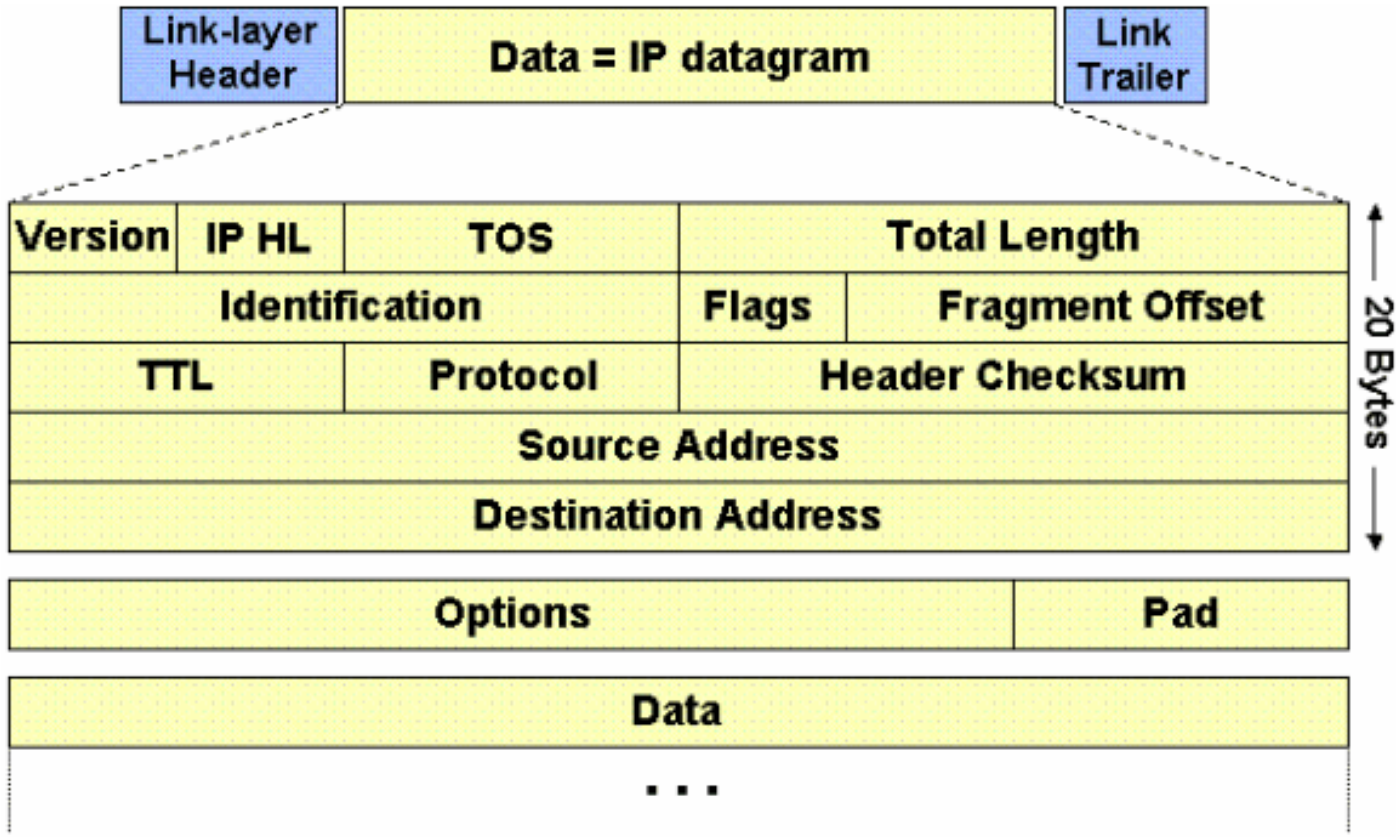
Fragmentation et réassemblage IP

Le protocole IP a été conçu en vue d'une utilisation sur une grande variété de liaisons de transmission. Bien que la longueur maximale d'un datagramme IP soit 65535, la plupart des liaisons de transmission imposent une plus petite limite maximum de longueur de paquet, appelée un MTU. La valeur de MTU dépend du type de liaison de transmission. La conception de l'IP

facilite des différences de MTU puisqu'elle permet à des Routeurs de fragmenter des datagrammes IP selon les besoins. La station de réception est responsable du réassemblage des fragments de nouveau dans le datagramme IP normal d'origine.

La fragmentation IP implique de diviser un datagramme en un certain nombre de parties qui peuvent être réassemblées plus tard. La source IP, la destination, l'identification, la longueur totale et les zones de décalage de fragment, ainsi que les indicateurs « autres fragments » et « ne pas fragmenter » figurant dans l'en-tête IP, sont utilisés pour la fragmentation et le réassemblage IP. Pour plus d'informations sur les mécanismes de la fragmentation et du réassemblage IP, voir le RFC 791 .

Cette image dépeint l'affichage d'une en-tête IP.



L'identification est 16 bits et est une valeur affectée par l'expéditeur d'un datagramme IP à faciliter le réassemblage des fragments d'un datagramme.

Le décalage de fragment est 13 bits et indique l'emplacement d'un fragment dans le datagramme IP d'origine. Cette valeur est un multiple de huit octets.

Dans le champ d'indicateurs de l'en-tête IP, il y a trois bits pour les indicateurs de contrôle. Il est important de noter que le bit « ne pas fragmenter » (DF) joue un rôle central dans PMTUD, parce qu'il détermine si un paquet peut être fragmenté.

0 mordu est réservé, et est toujours placé à 0. a mordu 1 est le bit DF (0 = « peut fragmenter, » 1 = « ne font pas fragment »). Le bit 2 représente le bit MF (0 = « dernier fragment », 1 = « autres fragments »).

Valeur	Bit 0 réservé	Bit 1 DF	Bit 2 MF
0	0	Mai	Dernier
1	0	Ne faites pas	Plus

Le prochain graphique affiche un exemple de fragmentation. Si vous ajoutez toutes les longueurs de fragments IP, la valeur dépasse de 60 la longueur du datagramme IP d'origine. La raison pour laquelle la longueur totale est augmentée de 60 est la suivante : trois en-têtes IP supplémentaires ont été créées (une pour chaque fragment suivant le premier fragment).

Le premier fragment a un décalage de 0, la longueur de ce fragment est 1500 ; cela inclut 20 octets pour l'en-tête IP d'origine légèrement modifiée.

Le deuxième fragment a un décalage de 185 (185 x 8 = 1480), ce qui signifie que la partie "données" de ce fragment démarre 1480 octets dans le datagramme IP d'origine. La longueur de ce fragment est 1500 ; cela inclut l'en-tête IP supplémentaire créée pour ce fragment.

Le troisième fragment a un décalage de 370 (370 x 8 = 2960), ce qui signifie que la partie "données" de ce fragment démarre 2960 octets dans le datagramme IP d'origine. La longueur de ce fragment est 1500 ; cela inclut l'en-tête IP supplémentaire créée pour ce fragment.

Le quatrième fragment a un décalage de 555 (555 x 8 = 4440), ce qui signifie que la partie "données" de ce fragment démarre 4440 octets dans le datagramme IP d'origine. La longueur de ce fragment est de 700 octets ; cela inclut l'en-tête IP supplémentaire créée pour ce fragment.

La taille du datagramme IP d'origine ne peut être déterminée qu'une fois le dernier fragment reçu.

Le décalage de fragment du dernier fragment (555) entraîne un décalage de données de 4440 octets dans le datagramme IP d'origine. Si vous ajoutez ensuite les octets de données du dernier fragment (680 = 700 - 20), on obtient 5120 octets, ce qui correspond à la partie "données" du datagramme IP d'origine. Ensuite, si vous ajoutez 20 octets pour une en-tête IP, vous obtenez la taille du datagramme IP d'origine (4440 + 680 + 20 = 5140).

Original IP Datagram

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

IP Fragments (Ethernet)

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

Problèmes de fragmentation IP

Il existe plusieurs problèmes qui rendent la fragmentation IP indésirable. On assiste à une faible augmentation de l'utilisation du processeur et de la mémoire pour la fragmentation d'un

datagramme IP. Cela se vérifie pour l'expéditeur aussi bien que pour le routeur dans le chemin compris entre un expéditeur et un récepteur. La création de fragments implique simplement de créer des en-têtes et de copier le datagramme initial dans les fragments. Cela peut être fait assez efficacement, car toutes les informations requises pour la création des fragments est immédiatement disponible.

La fragmentation entraîne davantage de surcharge pour le récepteur lors du réassemblage des fragments, car le récepteur doit allouer de la mémoire pour les fragments en entrée et les fusionner de nouveau dans un datagramme une fois tous les fragments reçus. Le réassemblage sur un hôte n'est pas considéré comme un problème, parce que l'hôte possède les ressources temporelles et de mémoire suffisantes pour l'exécution de cette tâche.

Mais le réassemblage est totalement inefficace sur un routeur dont le rôle principal consiste à expédier des paquets le plus rapidement possible. Les routeurs ne sont pas conçus pour la rétention de paquets pendant une durée indéterminée. Également un routeur qui fait le réassemblage choisit la plus grande mémoire tampon disponible (18K) avec laquelle pour travailler parce qu'elle n'a aucune manière de connaître la taille du paquet IP d'origine jusqu'à ce que le dernier fragment soit reçu.

Un autre problème de fragmentation implique comment des fragments abandonnés sont manipulés. Si un fragment de datagramme IP est supprimé, la totalité du datagramme IP d'origine doit être renvoyé, et il sera également fragmenté. Vous pouvez en voir un exemple avec le système NFS (Network File System). Le NFS, par défaut, a lue et écrit la longueur de bloc de 8192, ainsi un datagramme NFS IP/UDP sera approximativement 8500 octets (qui inclut des en-têtes NFS, d'UDP, et IP). Une station émettrice connectée à Ethernet (MTU 1500) devra fragmenter le datagramme de 8500 octets en six parties : cinq fragments de 1500 octets et un fragment de 1100 octets. Si l'un des six fragments sont abandonnés en raison d'une liaison encombrée, le datagramme d'origine complet devra être retransmis, ainsi il signifie que six davantage fragments devront être créés. Si cette liaison supprime l'un des six paquets, il y a peu de chances que les données NFS puissent être transférées sur cette liaison : en effet, un fragment IP minimum est dans ce cas supprimé de chaque datagramme IP d'origine (NFS 8500 octets).

Les Pare-feu qui filtrent ou manipulent des paquets basés sur la couche 4 (L4) par les informations de la couche 7 (L7) dans le paquet pourraient avoir le problème traitant des fragments IP correctement. Si les fragments IP sont en panne, un Pare-feu pourrait bloquer les fragments non initiaux parce qu'ils ne diffusent pas les informations qui apparieraient le filtre de paquet. Cela signifie que le datagramme IP d'origine ne peut pas être réassemblé par l'hôte de destination. Si le pare-feu est configuré de façon à autoriser les fragments non initiaux avec des informations insuffisantes en correspondance avec le filtre, une attaque de fragment non initial par le pare-feu risque de se produire. En outre, quelques paquets directs de périphériques de réseau (tels que les engines de commutateur satisfaites) basés sur L4 par les informations L7, et si un paquet répartit de plusieurs fragments, puis le périphérique pourraient avoir le problème imposant ses stratégies.

Évitez la fragmentation IP : Tâches et fonctionnement de TCP MSS

La taille maximum de segment TCP (MSS) définit la quantité maximale de données qu'un hôte souhaite accepter dans un datagramme TCP/IP simple. Ce datagramme TCP/IP pourrait être fragmenté à la couche IP. La valeur MSS est envoyée comme en-tête TCP uniquement dans les segments TCP SYN. Chaque côté d'une connexion TCP indique sa valeur MSS à l'autre côté. Contrairement à ce que l'on croit, la valeur MSS n'est pas négociée entre les hôtes. L'hôte expéditeur doit limiter la taille des données dans les segments TCP à une valeur inférieure ou

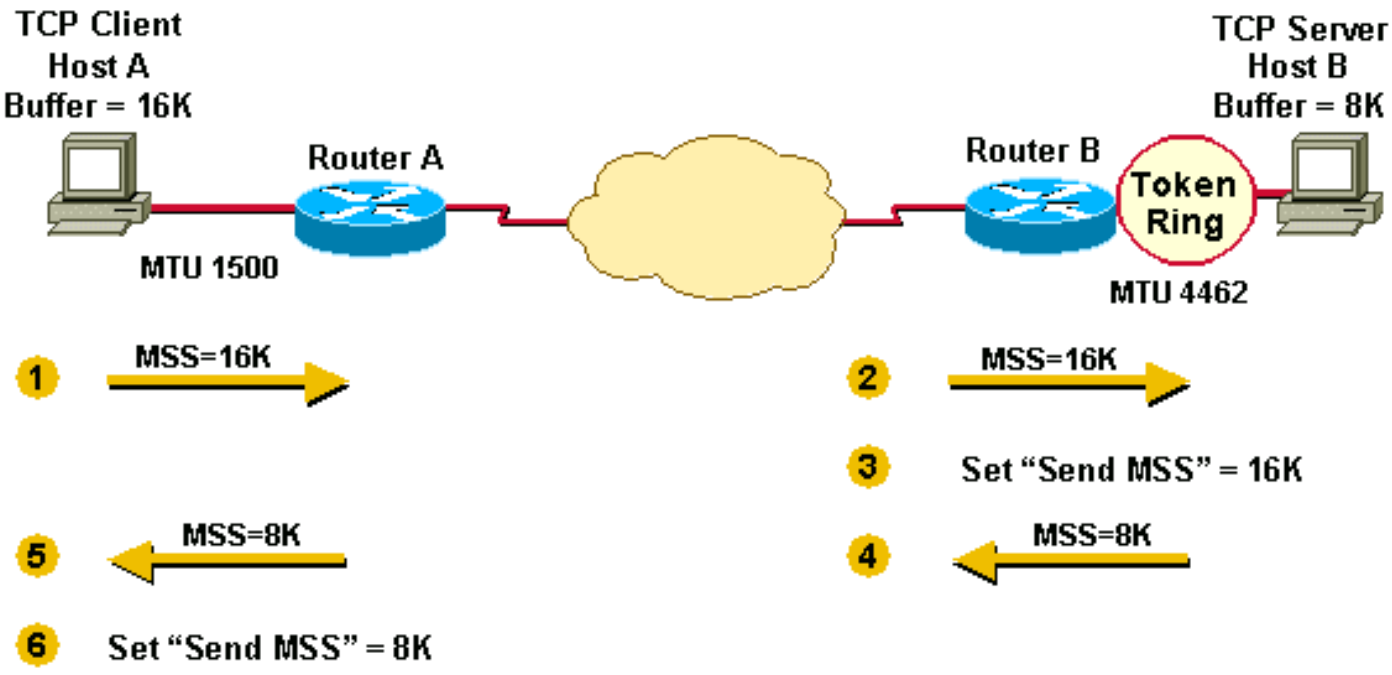
égale au MSS indiqué par l'hôte de destination.

Initialement, la valeur MSS indiquait la taille d'une mémoire tampon (supérieure ou égale à 65496K) allouée à une station de destination, lui permettant de stocker les données TCP contenues dans un datagramme IP simple. MSS était le segment maximum de données (bloc) que le récepteur TCP souhaitait accepter. Ce segment TCP pouvait atteindre 64K (la taille maximum de datagramme IP) et pouvait être fragmenté au niveau de la couche IP afin d'être transmis via le réseau à l'hôte de destination. L'hôte de destination pouvait alors réassembler le datagramme IP avant de transmettre le segment TCP complet à la couche TCP.

Sont ci-dessous quelques scénarios qui affichent comment des valeurs MSS sont placées et utilisées pour limiter des tailles de segment de TCP, et donc, des tailles de datagramme IP.

Le scénario 1 illustre la façon dont MSS a été mis en application la première fois. L'hôte A a une mémoire tampon de 16K et l'hôte B une mémoire tampon de 8K. Ils envoient et reçoivent leurs valeurs MSS et ajustent leur envoi MSS en vue de l'envoi réciproque de données. Notez que qui hébergent A et hôte B devra fragmenter les datagrammes IP qui sont plus grands que l'interface MTU, mais encore moins que l'envoi MSS parce que la pile de TCP pourrait passer les octets de données 16K ou 8K en bas de la pile à l'IP. Dans le cas de l'hôte B, les paquets ont été fragmentés deux fois (une fois pour l'accès au réseau local Token Ring et une deuxième fois pour l'accès au réseau local Ethernet).

Scénario 1



1. L'hôte A envoie sa valeur MSS de 16K à l'hôte B.
2. L'hôte B reçoit la valeur 16K MSS envoyée par l'hôte A.
3. L'hôte B définit sa valeur MSS à 16K.
4. L'hôte B envoie sa valeur MSS de 8K à l'hôte A.
5. L'hôte A reçoit la valeur 8K MSS envoyée par l'hôte A.
6. L'hôte A définit sa valeur MSS à 8K.

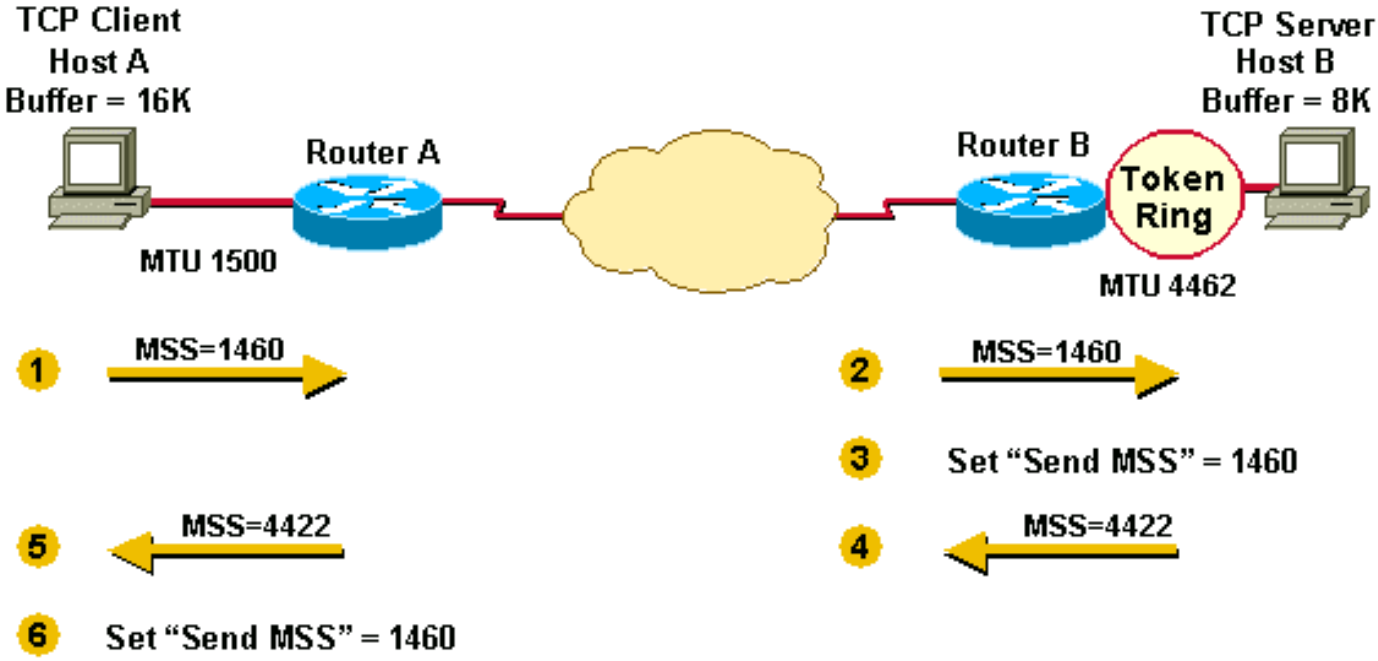
Afin d'éviter la fragmentation IP au niveau des points d'extrémité de la connexion TCP, la sélection de la valeur MSS a été modifiée ; elle est maintenant égale à la taille minimale de mémoire tampon et à la valeur MTU de l'interface sortante (- 40). Les valeurs MSS sont inférieures de 40

octets aux valeurs MTU, car MSS correspond à la taille de données TCP, ce qui n'inclut ni les 20 octets de l'en-tête IP, ni les 20 octets de l'en-tête TCP. MSS est basé sur les tailles d'en-tête par défaut ; la pile d'expéditeur doit soustraire les valeurs appropriées pour l'en-tête IP et la personne à charge d'en-tête de TCP en quelles options de TCP ou IP sont utilisées.

Le fonctionnement actuel de MSS est le suivant : chaque hôte compare tout d'abord son interface MTU sortante avec sa propre mémoire tampon, et choisit la valeur la plus faible en tant que valeur MSS à envoyer. Ensuite, les hôtes comparent la taille MSS reçue avec leur propre interface MTU, et choisissent à nouveau la plus faible des deux valeurs.

Le scénario 2 illustre cette étape supplémentaire prise par l'expéditeur afin d'éviter la fragmentation sur les fils locaux et distants. Vous remarquerez que la valeur MTU de l'interface sortante est prise en compte par chaque hôte (avant que les hôtes ne s'envoient mutuellement leurs valeurs MSS) et que cela permet d'éviter la fragmentation.

Scénario 2



1. L'hôte A compare sa mémoire tampon MSS (16K) et son MTU (1500 - 40 = 1460) et utilise la valeur la plus basse comme MSS (1460) pour l'envoyer à l'hôte B.
2. L'hôte B reçoit la valeur MSS envoyée par l'hôte A (1460) et la compare à la valeur MTU de son interface sortante - 40 (4422).
3. L'hôte B définit la valeur la plus basse (1460) comme valeur MSS utilisée pour l'envoi des datagrammes IP à l'hôte A.
4. L'hôte B compare sa mémoire tampon MSS (8K) et son MTU (4462-40 = 4422), puis utilise 4422 comme MSS à envoyer à l'hôte A.
5. L'hôte A reçoit la valeur MSS envoyée par l'hôte B (4422) et la compare à la valeur de son interface de sortie MTU - 40 (1460).
6. L'hôte A définit la valeur la plus basse (1460) comme valeur MSS utilisée pour l'envoi des datagrammes IP à l'hôte B.

1460 est la valeur choisie par les deux hôtes comme valeur MSS à s'envoyer réciproquement. Souvent la valeur de l'envoi MSS sera identique sur chaque extrémité d'une connexion TCP.

Dans le scénario 2, la fragmentation ne se produit pas au niveau des points d'extrémité d'une

connexion TCP, car les deux valeurs MTU de l'interface sortante sont prises en compte par les hôtes. Les paquets peuvent toutefois être fragmentés au sein du réseau entre le routeur A et le routeur B, s'ils trouvent une liaison portant une valeur MTU inférieure à celle de l'interface sortante des hôtes.

Qu'est-ce que PMTUD ?

Le TCP MSS comme plus tôt décrit prend soin de fragmentation aux deux points finaux d'une connexion TCP, mais elle ne traite pas le cas où il y a un plus petit lien de MTU au milieu entre ces deux points finaux. PMTUD a été développé afin d'éviter la fragmentation dans le chemin entre les points finaux. Il est utilisé pour déterminer dynamiquement la valeur MTU la plus faible entre la source et la destination d'un paquet.

> Remarque: PMTUD est seulement pris en charge par TCP et UDP. D'autres protocoles ne le prennent en charge pas. Si PMTUD est activé sur un hôte, et il est presque toujours, tous les TCP/IP ou paquets UDP de l'hôte aura le bit DF réglé.

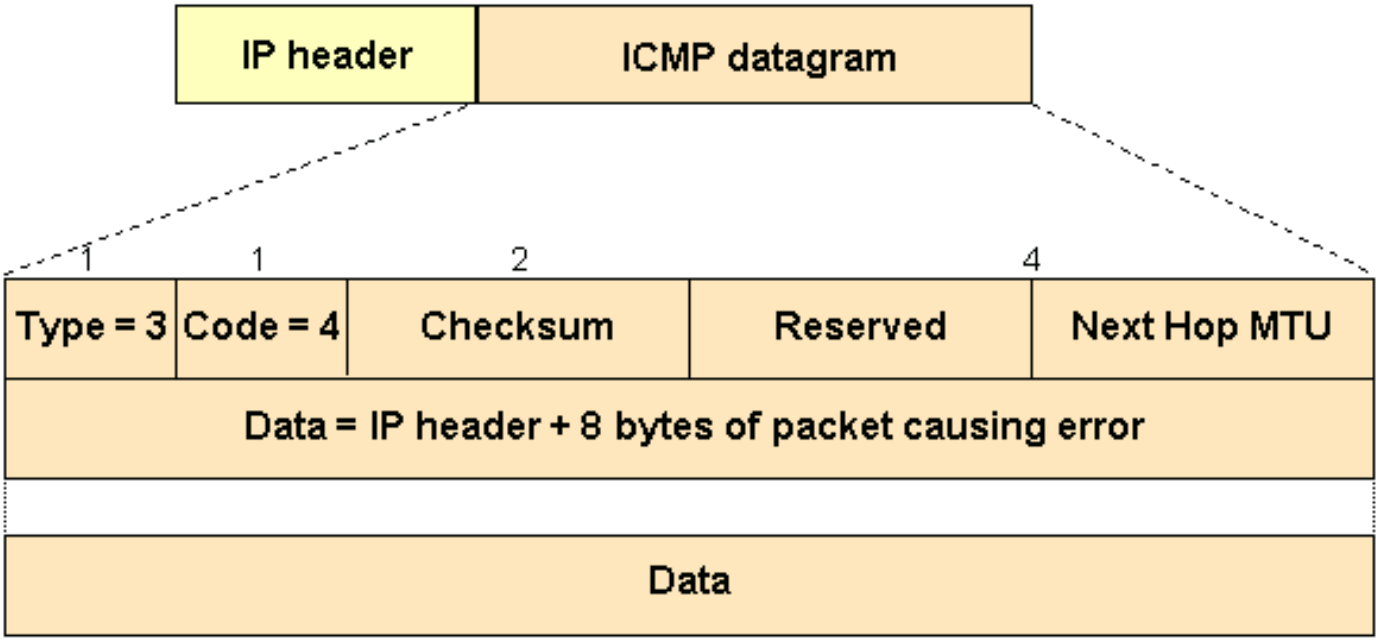
Quand un hôte envoie un plein paquet de données MSS avec le positionnement de bit DF, PMTUD réduit la valeur de l'envoi MSS pour la connexion s'il reçoit les informations que le paquet exigerait la fragmentation. Un hôte habituellement « se souvient » la valeur de MTU pour une destination puisqu'il crée une /32) entrée de « hôte » (dans sa table de routage avec cette valeur de MTU.

Si les essais d'un routeur pour expédier un datagramme IP, avec le positionnement de bit DF, sur un lien qui a un MTU inférieur que la taille du paquet, le routeur relâcheront le paquet et le message inaccessible de « destination renverront Protocole ICMP (Internet Control Message Protocol) » à la source de ce datagramme IP, avec le code qui indique « fragmentation requise et le DF placer » (type 3, code 4). Lorsque la station source reçoit le message ICMP, elle diminue la valeur MSS ; quand TCP retransmet le segment, il utilisera la taille de segment la plus faible.

Voici un exemple d'un ICMP « fragmentation requise et DF pour placer » le message que vous pourriez voir sur un routeur après que la commande de **debug ip icmp** soit activée :

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

Ce diagramme affiche le format de l'en-tête d'ICMP d'une « fragmentation requise et du DF pour placer » le message inaccessible de « destination ».



Par RFC 1191 , un routeur qui retourne un message ICMP qui indique « fragmentation a eu besoin et le DF placer » devrait inclure le MTU de ce réseau du prochain saut dans les 16 bits d'ordre réduit du champ d'en-tête supplémentaire d'ICMP qui est étiqueté « inutilisé » dans le RFC 792 de spécification d'ICMP .

Les premières mises en oeuvre de RFC 1191 n'ont pas fourni les informations MTU de saut suivant. Même lorsque ces informations ont été fournies, certains hôtes les ignorent. Pour ces situations, RFC 1191 contient également un tableau qui présente les valeurs suggérées par lesquelles la valeur MTU doit être diminuée pendant le PMTUD. Il est utilisé par des hôtes afin d'arriver plus rapidement à une valeur raisonnable pour l'envoi MSS.

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

PMTUD est exécuté continuellement sur tous les paquets, car le chemin entre l'expéditeur et le destinataire peut changer dynamiquement. Chaque fois qu'un expéditeur reçoit un message ICMP « fragmentation impossible », il met à jour les informations de routage (dans lesquelles il stocke le PMTUD).

Deux situations peuvent se produire pendant le PMTUD :

- Le paquet peut être acheminé jusqu'au destinataire sans être fragmenté.Remarque: Pour qu'un routeur puisse protéger le processeur contre les attaques DoS, il limite le nombre de messages ICMP qu'il envoie à deux par seconde. Par conséquent, dans ce contexte, si vous avez un scénario de réseau dans lequel vous prévoyez que le routeur devrait répondre avec plus de deux messages ICMP (type = 3, code = 4) par seconde (peuvent être les différents hôtes), vous voudriez désactiver l'étranglement des messages ICMP avec l'**aucune commande d'interface de l'ip icmp rate-limit unreachable [DF]**.

- L'expéditeur peut recevoir des messages ICMP « fragmentation impossible » en provenance de sauts situés sur le chemin du destinataire.

PMTUD est exécuté indépendamment pour les deux directions d'un flux TCP. Il pourrait y avoir des cas où PMTUD dans une direction d'un écoulement déclenche une des stations d'extrémité pour diminuer l'envoi MSS et l'autre station d'extrémité garde l'original pour envoyer MSS parce qu'elle n'a jamais envoyé un datagramme IP assez grand pour déclencher PMTUD.

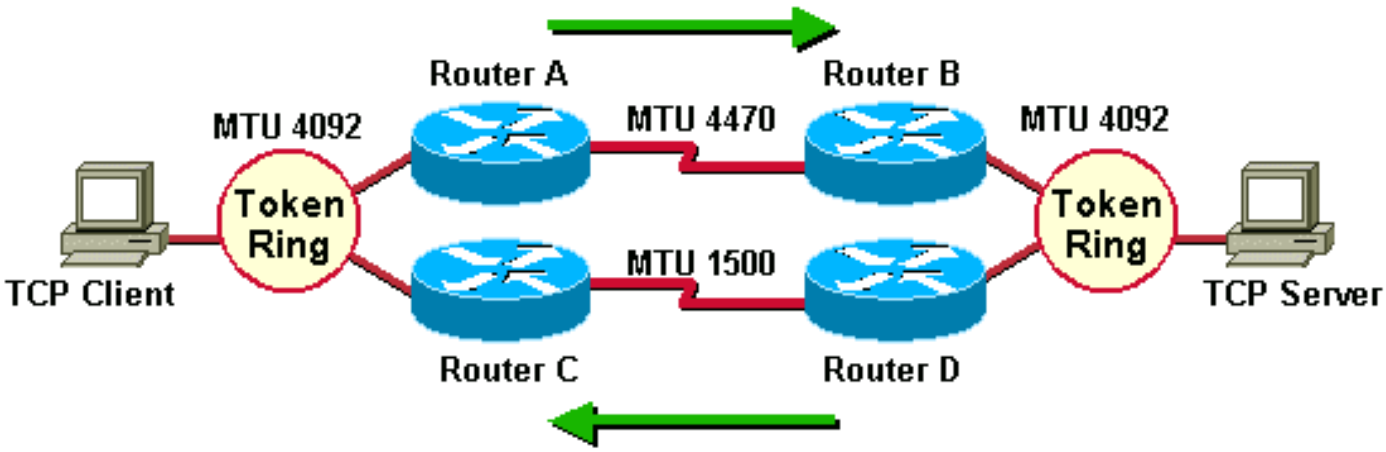
Vous trouverez un exemple de cette situation dans le scénario 3 (connexion HTTP). Le client de TCP envoie de petits paquets et le serveur envoie de grands paquets. Dans ce cas, seulement les grands paquets du serveur (plus considérablement que 576 octets) déclencheront PMTUD. Les paquets peu volumineux (inférieurs à 576 octets) ne déclenchent pas PMTUD, car ils n'exigent pas de fragmentation sur la liaison MTU 576.

Scénario 3

Le scénario 4 montre un exemple de routage asymétrique, dans lequel l'un des chemins affiche une valeur MTU minimale inférieure à celle de l'autre. Le routage asymétrique se produit quand des différents chemins sont pris pour envoyer et recevoir des données entre deux points finaux. Dans ce scénario, PMTUD déclenchera la diminution de la valeur MSS uniquement dans une direction de flux TCP. Le trafic du client de TCP au serveur traverse le routeur A et le routeur B, tandis que le trafic de retour qui provient le serveur au client traverse le routeur D et le routeur C. Quand le serveur TCP envoie des paquets au client, PMTUD déclenche le serveur afin de diminuer la valeur MSS, car le routeur D doit fragmenter les paquets de 4092 octets avant de pouvoir les envoyer au routeur C.

Le client, d'autre part, ne recevra jamais un ICMP message inaccessible de « destination » avec le code qui indique la « fragmentation requise et le DF pour placer » parce que le routeur A ne doit pas fragmenter des paquets quand il les envoie au serveur par le routeur B.

Scénario 4



Remarque: La commande **ip tcp path-mtu-discovery** est utilisée pour activer la découverte des voies d'accès MTU TCP pour les connexions TCP initialisées par des routeurs (BGP et Telnet par exemple).

Problèmes survenant avec la découverte PMTUD

Il existe trois problèmes potentiels au niveau de PMTUD, dont deux rares et un fréquent.

- Un routeur peut supprimer un paquet et ne pas envoyer de message ICMP. (Rare)
- Un routeur peut générer et envoyer un message ICMP, mais le message ICMP obtient bloqué par un routeur ou un Pare-feu entre ce routeur et l'expéditeur. (Terrain communal)
- Un routeur peut générer et envoyer un message ICMP, mais l'expéditeur peut ignorer ce message. (Rare)

Le premier et le dernier point de la liste ci-dessous correspondent à des situations rares et proviennent généralement d'une erreur, tandis que le point intermédiaire décrit un problème fréquent. Les personnes chargées de la mise en oeuvre de filtres de paquets ICMP ont tendance à bloquer tous les types de messages ICMP, au lieu de ne bloquer que certains d'entre eux. Un filtre de paquets peut bloquer tous les types de messages ICMP *à l'exception* de ceux qui indiquent « inaccessible » ou « dépassement de délai » Le succès ou l'échec de PMTUD est lié aux messages d'inaccessibilité ICMP parvenant à l'expéditeur d'un paquet TCP/IP. Les messages ICMP time-exceeded sont importants pour d'autres problèmes d'IP. Un exemple d'un tel filtre de paquet, mis en application sur un routeur est affiché ici.

```
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
```

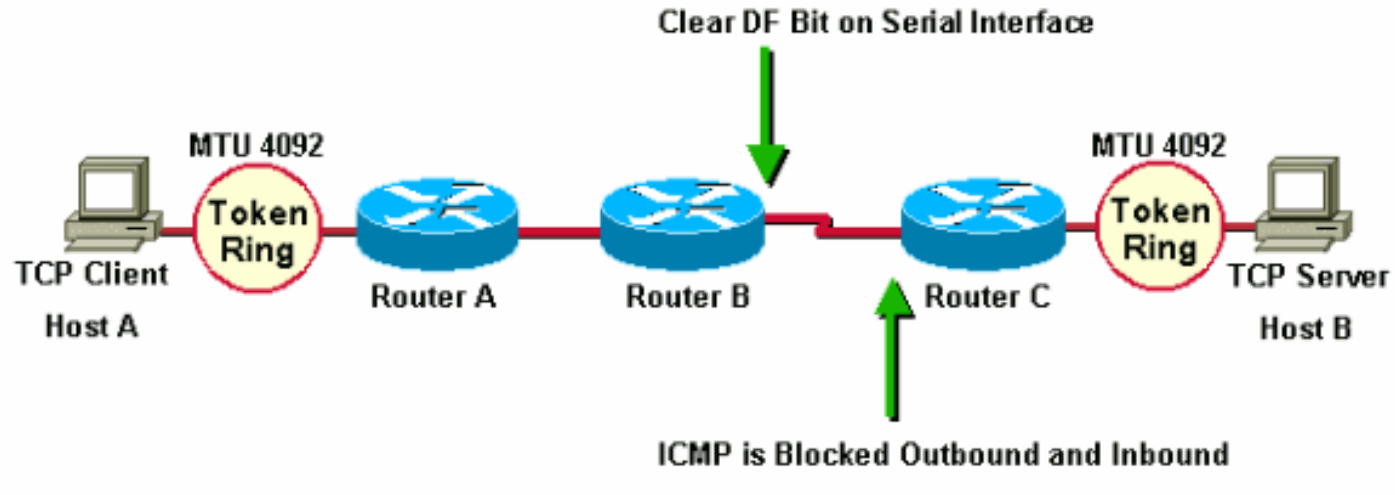
Il y a d'autres techniques qui peuvent être utilisées pour alléger le problème de blocage total de l'ICMP.

- Effacez le bit DF sur le routeur et permettez la fragmentation de toute façon (ceci ne pourrait pas être une bonne idée, cependant. Pour plus d'informations, voir Problèmes de fragmentation IP .
- Manipulez la valeur de l'option MSS de TCP MSS avec l'**ip tcp adjust-mss <500-1460> de** commande d'interface.

Dans le prochain scénario, le routeur A et le routeur B sont dans le même domaine administratif. Le routeur C est inaccessible et bloque l'ICMP, ainsi PMTUD est cassé. Un contournement pour cette situation est d'effacer le bit DF dans les deux directions sur le routeur B afin de permettre la fragmentation. Ceci peut être fait avec le routage de stratégie. La syntaxe à utiliser pour effacer le bit DF est disponible dans Cisco IOS® 12.1(6) et versions ultérieures.

```
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
```



Une autre possibilité consisterait à changer la valeur de l'option MSS TCP sur les paquets SYN qui traversent le routeur (disponible dans Cisco IOS 12.2(4)T et versions ultérieures). Ceci réduit la valeur de l'option MSS dans le paquet de synchronisation de TCP de sorte qu'il soit plus petit que la valeur (1460) dans la commande d'**ip tcp adjust-mss**. Le résultat est que l'expéditeur TCP enverra des segments qui ne dépasseront pas cette valeur. La taille de paquets IP sera 40 octets plus grande (1500) que la valeur MSS (1460 octets) afin d'expliquer l'en-tête de TCP (20 octets) et l'en-tête IP (20 octets).

Vous pouvez ajuster le MSS des paquets SYN TCP à l'aide de la commande **ip tcp adjust-mss** . Cette syntaxe ramènera la valeur MSS sur des segments de TCP à 1460. Cette commande permet de traiter le trafic entrant et sortant sur l'interface serial0.

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

Les problèmes de fragmentation IP sont devenus plus répandus depuis le déploiement massif des tunnels IP. La raison pour laquelle les tunnels entraînent plus de fragmentation est parce que l'encapsulation de tunnel ajoute le « temps système » à la taille d'un paquet. Par exemple, l'ajout de l'encapsulation générique de routeur (GRE) ajoute 24 octets à un paquet, et après que cette augmentation que le paquet pourrait avoir besoin pour être fragmenté parce qu'il est plus grand que le MTU sortant. Dans une section ultérieure de ce document, vous verrez des exemples de problèmes qui peuvent se poser avec les tunnels et la fragmentation IP.

Topologies réseau courantes nécessitant la découverte PMTUD

PMTUD est nécessaire dans les situations réseau dans lesquelles des liaisons intermédiaires possèdent des MTU plus faibles que le MTU des liaisons finales. Les raisons de ces valeurs plus faibles de MTU peuvent notamment être les suivantes :

- eToken Ring (ou FDDI) - hôtes d'extrémité connectés avec une connexion Ethernet entre elles. Les mtu d'Anneau à jeton (ou FDDI) aux extrémités sont plus grands que le MTU d'Ethernets au milieu.
- PPPoE (souvent utilisé avec ADSL) a besoin de 8 octets pour son en-tête. Ceci ramène le MTU efficace d'Ethernet à 1492 (1500 - 8).

Les protocoles de canalisation en tunnel comme GRE, IPsec, et L2TP ont besoin également d'espace pour leurs en-têtes et queues de bande respectifs. Cela réduit également le MTU efficace de l'interface sortante.

Dans les sections suivantes, l'incidence de PMTUD où un protocole de Tunnellisation est utilisé quelque part entre les deux hôtes d'extrémité sont étudiées. Des trois cas précédents, ce cas est le plus complexe et couvre toutes les questions que vous pourriez voir dans les autres cas.

Qu'est-ce qu'un tunnel ?

Un tunnel est une interface logique sur un routeur Cisco qui fournit une méthode d'encapsulation de paquets passagers au sein d'un protocole de transport. C'est une architecture conçue pour fournir les services destinés à mettre en application un plan d'encapsulation point à point. Le Tunnellisation a ces trois composants principaux :

- Protocole passager (AppleTalk, Banyan VINES, CLNS, DECNet, IP, ou IPX)
- Protocole transporteur - Un de ces protocoles d'encapsulation :GRE - Le protocole transporteur multiprotocole de Cisco. Voir RFC 2784 et RFC 1701 pour plus d'informations.IP dans des tunnels IP - Voir RFC 2003 pour plus d'informations.
- Protocole de transport - Protocol utilisé pour acheminer le protocole encapsulé

Les paquets affichés dans cette section illustrent les concepts de Tunnellisation IP où GRE est le protocole d'encapsulation et l'IP est le protocole de transport. Le protocole passager est également IP. Dans ce cas, IP représente à la fois le protocole de transport et le protocole passager.

Paquet normal

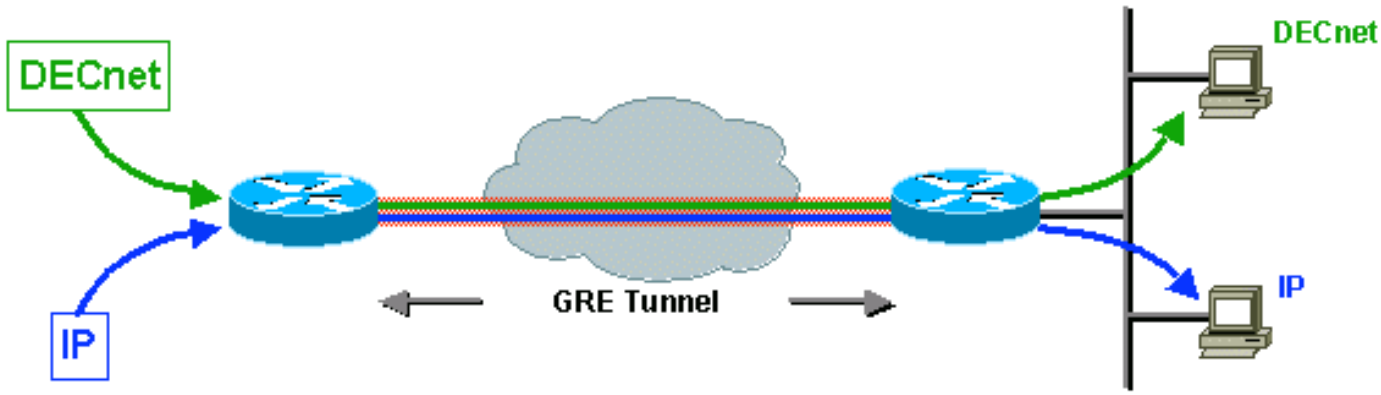
IP TCP Telnet

Paquet tunnel

IP GRE IP TCP Telnet

- IP est le protocole de transport.
- GRE est le protocole d'encapsulation.
- IP est le protocole passager.

L'exemple suivant montre l'encapsulation d'IP et de DECNet comme protocoles passagers avec GRE comme transporteur. Cela illustre le fait que le protocole transporteur peut encapsuler des protocoles passagers multiples.



Un administrateur réseau peut envisager la transmission tunnel dans une situation où deux réseaux non contigus et non IP sont séparés par un circuit principal IP. Si les réseaux discontinus exécutent le DECNet, l'administrateur ne pourrait pas vouloir les connecter ensemble en configurant le DECNet dans le circuit principal. L'administrateur ne pourrait pas vouloir permettre

au decnet routing pour consommer la bande passante principale parce que ceci pourrait gêner la représentation du réseau IP.

Une alternative viable est la transmission tunnel de DECNet sur le circuit principal IP. Le Tunnellisation encapsule les paquets de DECNet à l'intérieur de l'IP, et les envoie à travers le circuit principal au périphérique du tunnel où l'encapsulation est enlevée et les paquets de DECNet peuvent être conduits à leur destination par l'intermédiaire du DECNet.

Encapsuler le trafic à l'intérieur d'un autre protocole fournit ces avantages :

- Les points finaux utilisent les adresses privées (RFC 1918) et le circuit principal ne prend en charge pas conduire ces adresses.
- Autorisez les VPN (Virtual Private Network) sur les WAN ou sur Internet.
- Joignez les réseaux multiprotocole non contigus sur un circuit principal à protocole unique.
- Cryptez le trafic sur le circuit principal ou sur Internet.

Pour le reste du document, l'IP est utilisé comme protocole passager et IP comme protocole de transport.

Considérations concernant les interfaces de tunnel

Ce sont des considérations en perçant un tunnel.

- La commutation rapide des tunnels GRE a été introduite dans le logiciel Cisco IOS version 11.1 et la commutation CEF a été introduite dans la version 12.0. La commutation CEF pour les tunnels GRE multipoints a été introduite dans la version 12.2(8)T. L'encapsulation et le décapsulage aux périphériques du tunnel étaient des exécutions lentes dans les versions antérieures du Cisco IOS quand seulement la commutation de processus a été prise en charge.
- Lors de la transmission tunnel de paquets, des problèmes de sécurité et de topologie se posent. Les tunnels peuvent ignorer les listes de contrôle d'accès (ACL) et les pare-feu. Si vous effectuez une transmission tunnel via un pare-feu, vous évitez le pare-feu pour tout protocole passager impliqué dans la transmission tunnel. Par conséquent, il est recommandé d'inclure la fonction de pare-feu aux extrémités du tunnel, afin de mettre en oeuvre une politique au niveau des protocoles passagers.
- Le Tunnellisation pourrait créer des problèmes avec les protocoles de transport qui ont limité des temporisateurs (par exemple, DECNet) en raison de la latence accrue.
- Perçant un tunnel à travers des environnements avec différents liens de vitesse, comme les anneaux FDDI rapides et par les lignes téléphoniques 9600-bps lentes, pourrait introduire le paquet commandant à nouveau des problèmes. Certains protocoles passagers fonctionnent mal dans des réseaux de supports mixtes.
- Les tunnels point à point peuvent épuiser la bande passante sur une liaison physique. Si vous exécutez des protocoles de routage au-dessus de plusieurs tunnels point par point, maintenez dans l'esprit que chaque interface de tunnel a une bande passante et que l'interface physique au-dessus dont le tunnel fonctionne a une bande passante. Par exemple, vous pouvez définir la bande passante du tunnel à 100 Kb s'il y a 100 tunnels sur une liaison de 10 Mb. La bande passante par défaut pour un tunnel est 9Kb.
- Les protocoles de routage pourraient préférer un tunnel au-dessus d'un « vrai » lien parce que le tunnel pourrait trompeusement sembler être un lien d'un-saut avec le chemin le plus peu coûteux, bien qu'il réellement implique plus de sauts et soit vraiment plus coûteux qu'un autre

chemin. Cela peut être atténué avec la configuration appropriée du protocole de routage. Vous pouvez envisager d'exécuter un protocole de routage différent sur l'interface de tunnel que celui exécuté sur l'interface physique.
- Les problèmes de routage récursif peuvent être évités via la configuration appropriée de routes statiques vers la destination du tunnel. On parle de route récursive quand le meilleur chemin vers la destination du tunnel récursif est le tunnel lui-même. Cette situation fait rebondir l'interface de tunnel en haut et en bas. Vous verrez cette erreur quand il y a un problème récursif de routage.`%TUN-RECURDOWN Interface Tunnel 0 temporarily disabled due to recursive routing`

Le routeur possède un participant PMTUD au niveau du point d'extrémité d'un tunnel

Le routeur a deux rôles PMTUD différents à jouer lorsqu'il représente le point d'extrémité d'un tunnel.

- Dans le premier rôle, le routeur est l'expéditeur d'un paquet hôte. Pour le traitement PMTUD, le routeur doit contrôler le bit DF et la taille de paquet du paquet de données d'origine, puis exécuter l'action appropriée, le cas échéant.
- Le deuxième rôle intervient une fois que le routeur a encapsulé le paquet IP d'origine dans le paquet de tunnel. À ce stade, le routeur agit plutôt un hôte en ce qui concerne PMTUD et en vue de le paquet IP de tunnel.

Permet le début en regardant ce qui se produit quand le routeur agit dans le premier rôle, un routeur qui hébergent en avant des paquets IP, en ce qui concerne PMTUD. Ce rôle intervient avant l'encapsulation du paquet IP hôte par le routeur dans le paquet du tunnel.

Si le routeur participe car l'expéditeur d'un paquet d'hôte il se terminera ces actions :

- Vérifier si le bit DF a été défini.
- Vérifier la taille de paquet utilisable par le tunnel.
- Fragmenter (si le paquet est trop volumineux et si le bit DF n'est pas défini), encapsuler et envoyer les fragments ; ou
- Supprimer le paquet (si le paquet est trop volumineux et si le bit DF est défini) et envoyer un message ICMP à l'expéditeur.
- Effectuer l'encapsulation (si le paquet n'est pas trop volumineux) et l'envoi.

Génériquement, il y a un choix de l'encapsulation et puis fragmentation (envoyez deux fragments d'encapsulation) ou fragmentation et puis d'encapsulation (envoyez deux fragments encapsulés).

Quelques exemples qui décrivent les mécanismes de l'encapsulation et de la fragmentation de paquet IP et deux scénarios qui affichent l'interaction de PMTUD et les paquets que des réseaux d'exemple transversaux sont détaillé dans cette section.

Le premier exemple affiche ce qui arrive à un paquet quand le routeur (à la source du tunnel) agit dans le rôle du routeur d'expédition. Souvenez-vous cela pour traiter PMTUD, le routeur doit vérifier le bit DF et la longueur de paquet du paquet de données d'origine et agir la mesure appropriée. Ces exemples utilisent l'encapsulation GRE pour le tunnel. Comme peut être vu, GRE fait la fragmentation avant l'encapsulation. Les exemples suivants montrent les scénarios dans lesquels la fragmentation est faite après l'encapsulation.

Dans l'exemple 1, le bit DF n'est pas défini (DF = 0) et le tunnel GRE IP MTU est 1476 (1500 -

24).

Exemple 1

1. Le routeur transporteur (à la source du tunnel) reçoit un datagramme de 1500 octets avec bit DF envoyé (DF = 0) en provenance de l'hôte expéditeur. Ce datagramme se compose d'un en-tête IP de 20 octets et d'une charge utile TCP de 1480 octets.
2. Puisque le paquet sera trop volumineux pour le MTU IP après ajout du temps système GRE (24 octets), le routeur transporteur divise le datagramme en deux fragments de 1476 (en-tête IP de 20 octets + charge utile IP de 1456 octets) plus 44 octets (20 octets d'en-tête IP + 24 octets de charge utile IP) ; par conséquent, une fois l'encapsulation GRE ajoutée, le paquet n'est pas plus volumineux que le MTU de l'interface physique sortante.
3. Le routeur transporteur ajoute l'encapsulation GRE (qui inclut un en-tête GRE de 4 octets et un en-tête IP de 20 octets) à chaque fragment du datagramme IP d'origine. Ces deux datagrammes IP ont maintenant une longueur de 1500 et 68 octets et ces datagrammes sont vus en tant que différents datagrammes IP, pas en tant que fragments.
4. Le routeur de destination de tunnel enlève l'encapsulation GRE de chaque fragment du datagramme d'origine, qui laisse deux fragments IP des longueurs 1476 et 24 octets. Ces fragments de datagrammes IP seront expédiés séparément par ce routeur au hôte de destination.
5. L'hôte de destination effectuera le réassemblage de ces deux fragments pour reconstituer le datagramme d'origine.

Le scénario 5 dépeint le rôle du routeur transporteur dans le contexte d'une topologie réseau.

Dans cet exemple le routeur agit dans le même rôle du routeur d'expédition, mais cette fois le bit DF est placé (DF = 1).

Exemple 2

1. Le routeur transporteur (à la source du tunnel) reçoit un datagramme de 1500 octets avec bit DF= 1) en provenance de l'hôte expéditeur.
2. Puisque le bit DF est défini et que la taille du datagramme (1500 octets) est plus grande que le MTU IP du tunnel GRE (1476), le routeur supprimera le datagramme et enverra un message à la source du datagramme, indiquant que la fragmentation ICMP est requise mais que le bit DF est défini. Le message ICMP alertera l'expéditeur que le MTU est 1476.
3. L'hôte expéditeur reçoit le message ICMP, et quand il renvoie les données d'origine il utilisera un datagramme IP 1476-byte.
4. Cette longueur de datagramme IP (1476 octets) est maintenant égale à la valeur du MTU IP du tunnel GRE. Par conséquent, le routeur ajoute l'encapsulation GRE au datagramme IP.
5. Le routeur de destination (à la destination du tunnel) supprime l'encapsulation GRE du datagramme IP et l'envoie à l'hôte destinataire.

Maintenant nous pouvons les regarder ce qui se produit quand le routeur agit dans le deuxième rôle comme hôte expéditeur en ce qui concerne PMTUD et en vue de le paquet IP de tunnel. Rappelez-vous que ce rôle intervient une fois que le routeur a encapsulé le paquet IP d'origine dans le paquet de tunnel.

> Remarque: Par défaut un routeur ne fait pas PMTUD sur les paquets de tunnel GRE qu'il génère. La commande **tunnel path-mtu-discovery** commande peut être utilisée pour activer PMTUD pour les paquets de tunnel GRE-IP.

L'exemple 3 affiche ce qui se produit quand l'hôte envoie les datagrammes IP qui sont assez petits pour entrer dans l'IP MTU sur l'interface de tunnel GRE. Dans ce cas, le bit DF peut être soit défini, soit effacé (1 ou 0). L'interface de tunnel GRE n'a pas la commande de **tunnel path-mtu-discovery** a configuré ainsi le routeur ne fera pas PMTUD sur le paquet GRE-IP.

Exemple 3

1. Le routeur transporteur (à la source du tunnel) reçoit un datagramme de 1476 octets en provenance de l'hôte expéditeur.
2. Ce routeur encapsule le datagramme IP de 1476 octets à l'intérieur de GRE afin d'obtenir un datagramme GRE IP de 1500 octets. Le bit DF de l'en-tête GRE IP est effacé (DF = 0). Ce routeur achemine ensuite ce paquet vers la destination du tunnel.
3. Supposons qu'il y ait un routeur entre la source et la destination du tunnel, avec un MTU de liaison de 1400. Ce routeur fragmentera le paquet du tunnel, car le bit DF est effacé (DF = 0). Rappelez-vous que dans cet exemple, l'IP extérieur est fragmenté, ainsi que le GRE, et que les en-têtes TCP ne sont affichées que dans le premier fragment.
4. Le routeur de destination du tunnel doit rassembler le paquet de tunnel GRE.
5. Une fois le paquet de tunnel GRE réassemblé, le routeur retire l'en-tête IP GRE et envoie le datagramme IP d'origine à sa destination.

L'exemple suivant affiche ce qui se produit quand le routeur agit dans le rôle d'un hôte expéditeur en ce qui concerne PMTUD et en vue de le paquet IP de tunnel. Cette fois le bit DF est placé (le DF = 1) dans l'en-tête IP d'origine et la commande de **tunnel path-mtu-discovery** a été configuré de sorte que le bit DF soit copié de l'en-tête IP intérieure sur (GRE + IP) l'en-tête externe.

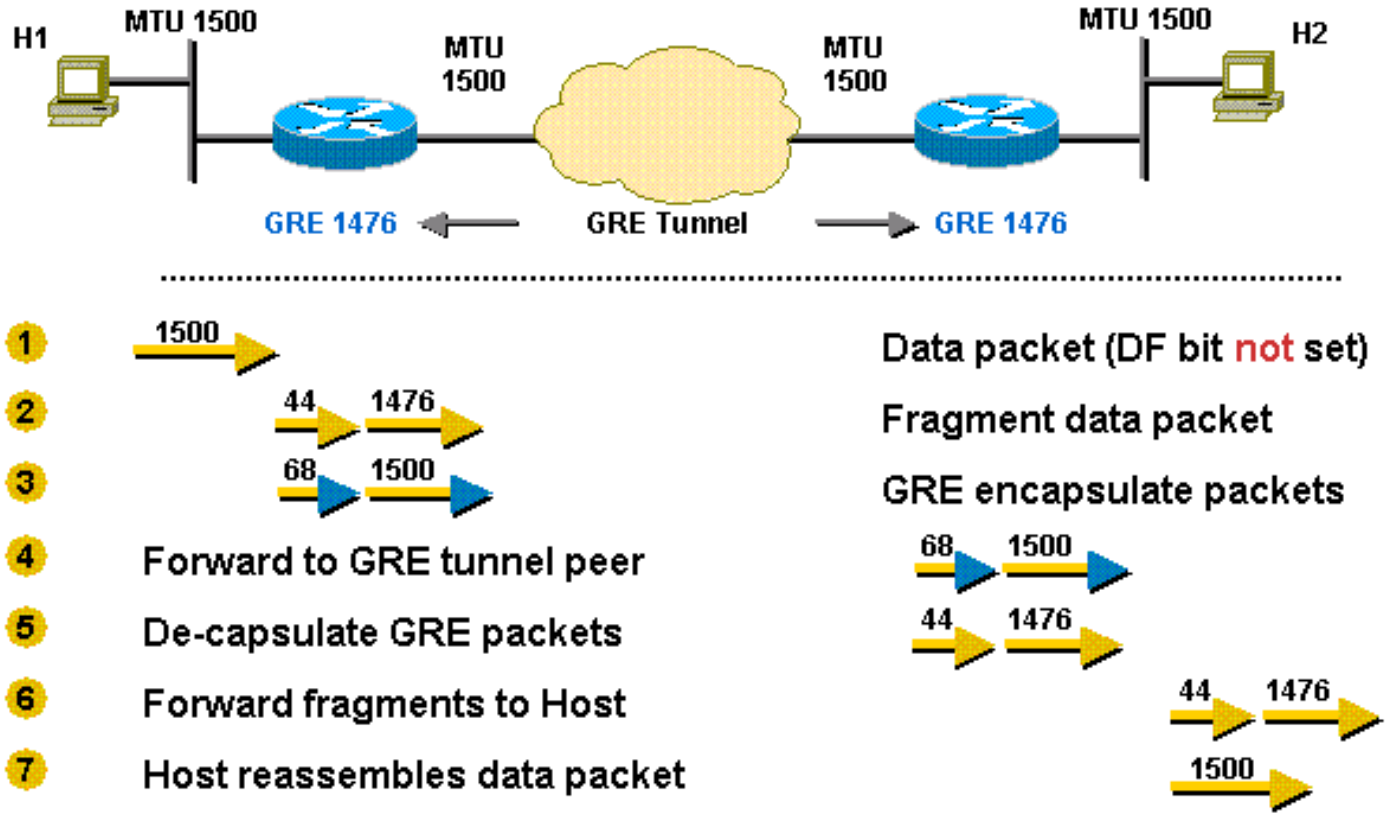
Exemple 4

1. Le routeur transporteur (à la source du tunnel) reçoit un datagramme de 1476 octets avec bit DF= 1) en provenance de l'hôte expéditeur.
2. Ce routeur encapsule le datagramme IP de 1476 octets à l'intérieur de GRE afin d'obtenir un datagramme GRE IP de 1500 octets. Cette en-tête IP GRE porte le bit DF défini (DF = 1) puisque le datagramme IP d'origine le portait également. Ce routeur achemine ensuite ce paquet vers la destination du tunnel.
3. Supposons de nouveau qu'il y ait un routeur entre la source et la destination du tunnel, avec un MTU de liaison de 1400. Ce routeur ne fragmentera pas le paquet de tunnel, car le bit DF est défini (DF = 1). Ce routeur doit supprimer le paquet et envoyer un message d'erreur ICMP au routeur source du tunnel, puisque c'est l'adresse IP de la source sur le paquet.
4. Le routeur transporteur (à la source du tunnel) reçoit ce message d'erreur ICMP et diminue le MTU IP du tunnel GRE en faisant passer sa valeur à 1376 (1400 - 24). La prochaine fois que l'hôte expéditeur retransmettra les données au sein d'un paquet IP de 1476 octets, ce paquet sera trop volumineux et ce routeur enverra un message d'erreur ICMP à l'expéditeur, portant la valeur MTU de 1376. Quand l'hôte expéditeur retransmet les données, il envoie un paquet IP de 1376 octets, et ce paquet est acheminé via le tunnel GRE jusqu'à l'hôte destinataire.

Scénario 5

Ce scénario illustre la fragmentation GRE. Rappelez-vous que vous devez fragmenter avant l'encapsulation pour GRE, puis que vous devez appliquer PMTUD pour le paquet de données ; notez également que le bit DF n'est pas copié lorsque le paquet IP est encapsulé par GRE. Dans

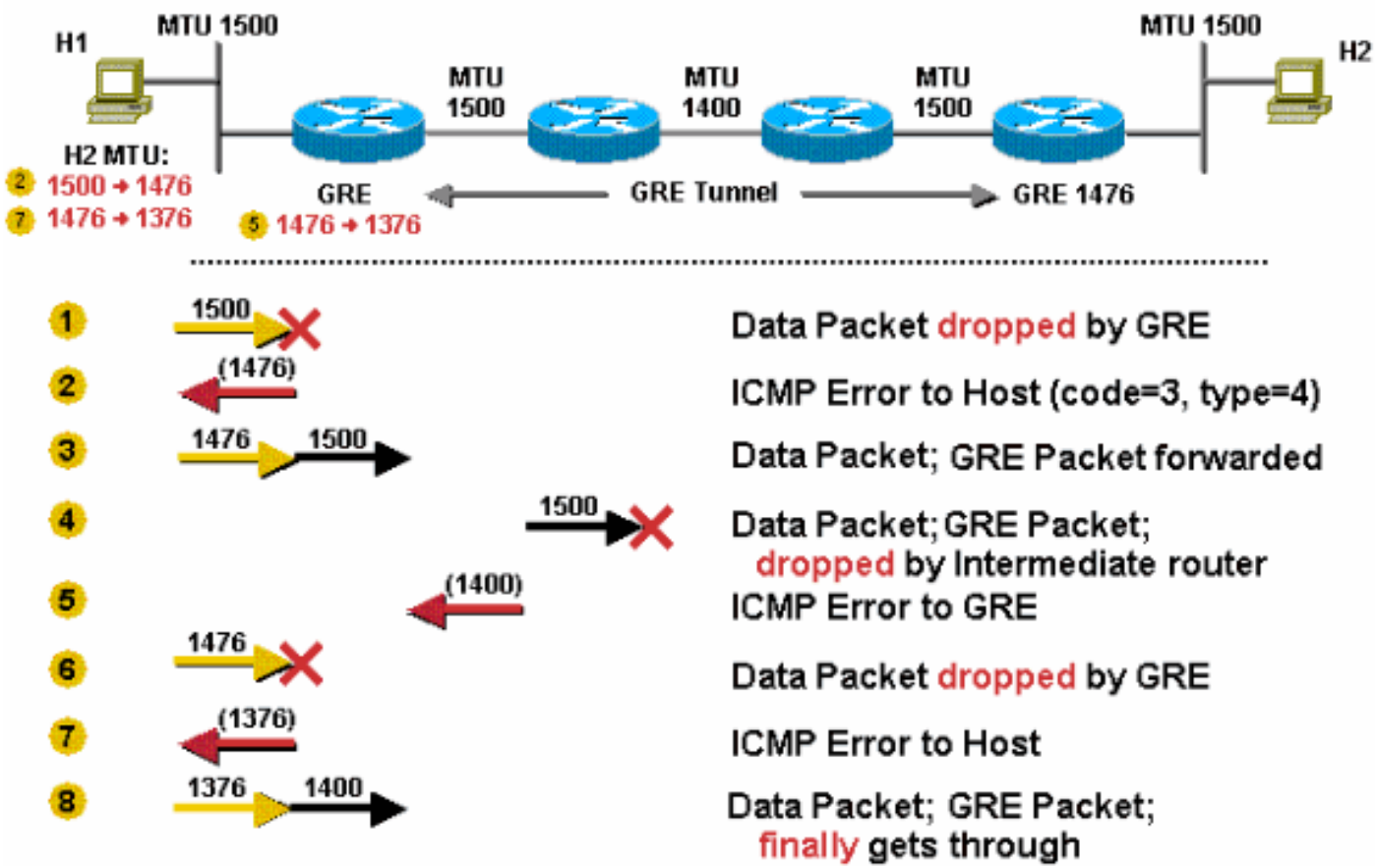
ce scénario, le bit DF n'est pas défini. Le MPU IP de l'interface de tunnel GRE est par défaut de 24 octets, soit inférieur au MTU IP de l'interface physique. Le MTU IP de l'interface GRE est donc de 1476.



1. L'expéditeur envoie un paquet 1500-byte (20 en-têtes IP d'octet + 1480 octets de la charge utile de TCP).
2. Puisque le MTU du tunnel GRE est 1476, le paquet de 1500 octets est divisé en deux fragments IP de 1476 et 44 octets, chacun en prévision des 24 octets supplémentaires de l'en-tête GRE.
3. Les 24 octets de l'en-tête GRE sont ajoutés à chaque fragment IP. Maintenant les fragments sont de 1500 (1476 + 24) et 68 (44 + 24) octets chacun.
4. GRE + paquets IP qui contiennent les deux fragments IP sont expédiés au routeur de pair de tunnel GRE.
5. Le routeur partenaire de tunnel GRE retire les en-têtes GRE des deux packets.
6. Ce routeur achemine les deux paquets vers l'hôte de destination.
7. L'hôte de destination réassemble les fragments IP dans le datagramme IP d'origine.

Scénario 6

Ce scénario est semblable au scénario 5, mais cette fois le bit DF est placé. Dans le scénario 6, le routeur est configuré de façon à appliquer PMTUD sur GRE + sur les paquets IP de tunnel à l'aide de la commande **tunnel path-mtu-discovery** , et le bit DF est copié de l'en-tête d'origine vers l'en-tête IP GRE. Si le routeur reçoit une erreur ICMP pour le GRE + paquet IP, il diminue la valeur MTU IP sur l'interface de tunnel GRE. De nouveau, rappelez-vous que le tunnel MTU IP du tunnel GRE est égal à 24 octets, soit moins que le MTU de l'interface physique par défaut ; par conséquent, dans ce cas le MTU IP GRE est égal à 1476. Notez également qu'il y a une liaison MTU 1400 dans le chemin du tunnel GRE.



1. Le routeur reçoit un paquet de 1500 octets (en-tête IP de 20 octets + charge utile TCP de 1480), et il supprime le paquet. Le routeur relâche le paquet parce qu'il est plus grand que l'IP MTU (1476) sur l'interface de tunnel GRE.
2. Le routeur envoie une erreur ICMP à l'expéditeur lui indiquant que le MTU du saut suivant est de 1476. L'hôte enregistre ces informations, habituellement sous forme de route hôte de destination dans sa table de routage.
3. L'hôte expéditeur utilise une taille de paquet de 1476 octets lorsqu'il envoie de nouveau les données. Le routeur GRE ajoute 24 octets d'encapsulation GRE et expédie un paquet de 1500 octets.
4. Le paquet de 1500 octets ne peut pas traverser la liaison de 1400 octets, il est donc supprimé par le routeur intermédiaire.
5. Le routeur intermédiaire envoie un ICMP (type = 3, code = 4) au routeur GRE avec un MTU de prochain-saut de 1400. Le routeur GRE fait passer cette valeur à 1376 (1400 - 24) et définit une valeur interne de MTU IP MTU sur l'interface GRE. Cette modification peut seulement être vue en utilisant la **commande debug tunnel** ; Elle ne peut pas être observée dans la sortie de la commande **show ip interface tunnel<#>** .
6. La prochaine fois l'hôte renvoie le paquet 1476-byte, le routeur GRE relâchera le paquet, puisqu'il est plus grand que l'IP MTU en cours (1376) sur l'interface de tunnel GRE.
7. Le routeur GRE enverra un autre ICMP (le type = 3, code = 4) à l'expéditeur avec un MTU de prochain-saut de 1376 et l'hôte mettra à jour ses informations en cours avec la nouvelle valeur.
8. L'hôte envoie de nouveau les données, mais cette fois-ci dans un paquet plus petit de 1376 octets, et GRE ajoute 24 octets d'encapsulation, puis l'achemine. Cette fois le paquet le fera au pair de tunnel GRE, où le paquet sera désencapsulé et envoyé à la destination host.Remarque: Si la commande **tunnel path-mtu-discovery** n'était pas configurée sur le routeur transporteur dans ce scénario, et si le bit DF était défini dans les paquets envoyés

via le tunnel GRE, l'hôte 1 parviendrait malgré tout à envoyer les paquets TCP/IP à l'hôte 2, mais en les fragmentant au milieu, au niveau de la liaison MTU 1400. Le partenaire de tunnel GRE devrait également les réassembler avant de pouvoir supprimer l'encapsulation et les acheminer.

"Mode Tunnel IPsec « pur »

La sécurité IP (IPsec) Protocol est une méthode basée sur des standards qui fournit l'intimité, l'intégrité, et l'authenticité aux informations transférées à travers des réseaux IP. IPsec fournit le cryptage de la couche réseau IP. IPsec rallonge le paquet IP en ajoutant au moins une en-tête IP (mode tunnel). L'en-tête ajoutée varie dans la personne à charge de longueur sur le mode de configuration d'IPsec mais elles ne dépassent pas ~58 octets (Protocole ESP (Encapsulating Security Payload) et authentification de l'ESP (ESPauth)) par paquet.

IPsec a deux modes, le mode tunnel et le mode transport.

- Le mode tunnel est le mode par défaut. Avec le mode tunnel, le paquet IP initial entier est protégé (crypté, authentifié, ou les deux) et encapsulé par les en-têtes et les queues de bande IPsec. Alors une nouvelle en-tête IP est ajoutée au début au paquet, qui des specifes les points finaux d'IPsec (pairs) comme source et destination. Le mode tunnel peut être utilisé avec tout trafic IP unicast et doit être utilisé si IPsec protège le trafic des hôtes derrière les homologues IPsec. Par exemple, le mode tunnel est utilisé avec des réseaux virtuels privés (VPN) (vpn) sur lesquels les hôtes d'un réseau protégé envoient des paquets aux hôtes d'un autre réseau protégé, via une paire d'homologues IPsec. Avec les VPN, le tunnel IPsec protège le trafic IP entre les hôtes, via le cryptage de ce trafic entre les routeurs homologues IPsec.
- Avec le mode transport (configuré avec la commande secondaire **mode transport** dans la définition de transformation), seule la charge utile du paquet IP initial est protégée (cryptée, authentifiée, ou les deux). La charge utile est encapsulée par les en-têtes et les remorques d'IPsec. Les en-têtes initiaux d'IP restent intacts, à l'exception de la zone de protocole IP qui est modifiée en ESP (50), et la valeur de protocole d'origine est enregistrée dans la queue de bande IPsec en vue d'une restauration une fois le paquet décrypté. Le mode transport est utilisé seulement quand le trafic IP à protéger est entre les homologues IPsec eux-mêmes, et que les adresses IP source et de destination sont les mêmes que les adresses d'homologues IPsec. Normalement, le mode transport IPsec transport mode est seulement utilisé quand un autre protocole de canalisation en tunnel (comme GRE) est utilisé d'abord pour encapsuler le paquet de données IP, puis IPsec est utilisé pour protéger les paquets de tunnel GRE.

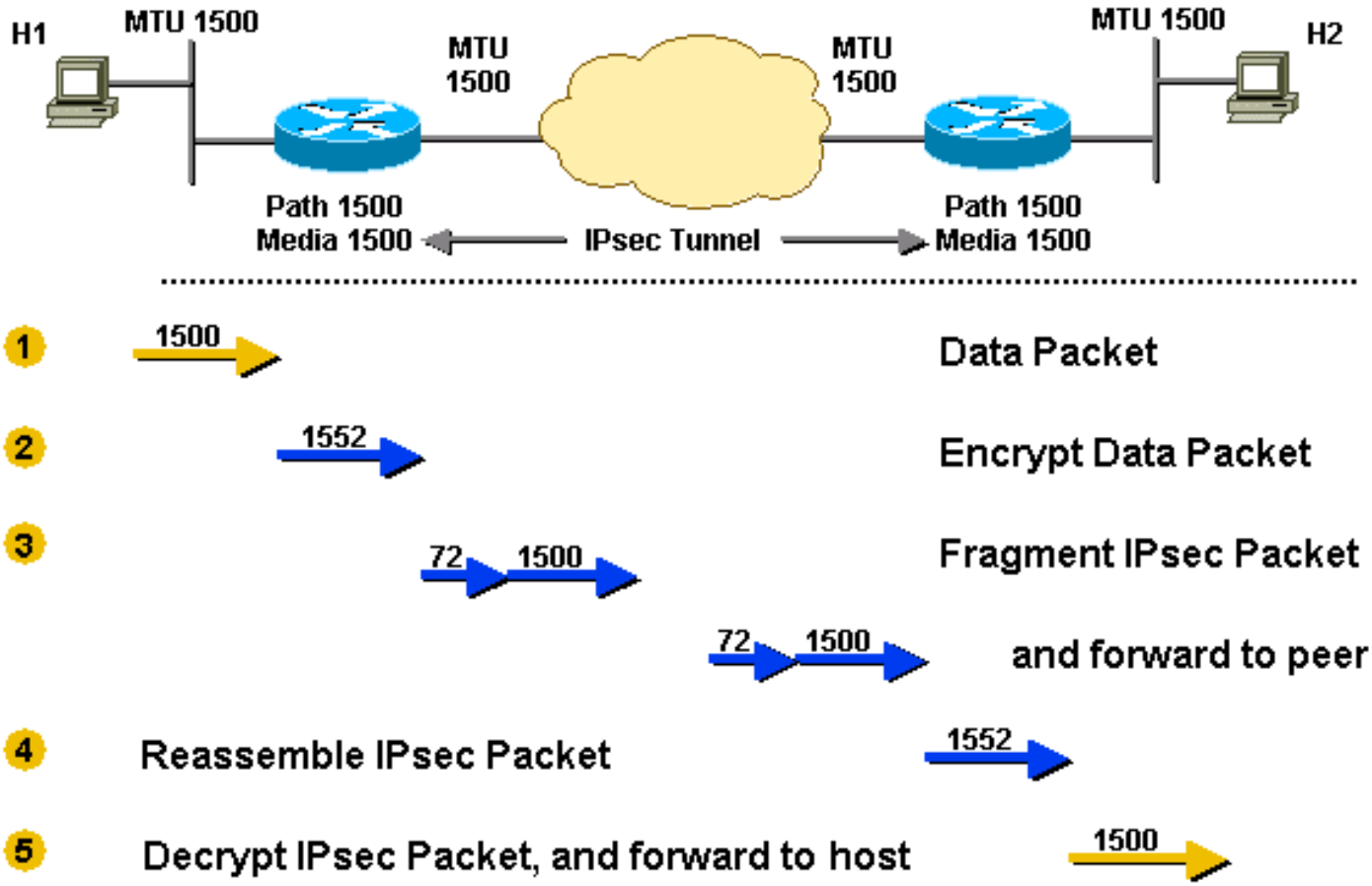
IPsec utilise toujours PMTUD pour les paquets de données et pour ses propres paquets. Certaines commandes de configuration IPsec permettent de modifier le traitement PMTUD pour le paquet IPsec IP, IPsec peut effacer, définir ou copier le bit DF à partir de l'en-tête IP de paquet de données dans l'en-tête IP d'IPsec. Ceci s'appelle « la fonction de remplacement de bit DF ».

> Remarque: Vous souhaitez éviter la fragmentation après l'encapsulation quand vous faites un cryptage matériel avec IPsec. Le cryptage matériel peut permettre d'obtenir un débit d'environ 50 Mbs, en fonction du matériel, mais si le paquet IPsec est fragmenté, vous perdez de 50 à 90 pour cent du débit. Cette perte s'explique par le fait que les paquets IPsec fragmentés sont commutés pour le réassemblage, puis transmis au moteur de chiffrement, à des fins de décryptement. Cette perte de débit peut mener le débit de cryptage matériel au

niveau des performances de cryptage logiciel (2-10 Mbs).

Scénario 7

Ce scénario décrit la fragmentation IPsec. Dans ce scénario, le MTU du chemin entier est 1500. Dans ce scénario, le bit DF n'est pas défini.



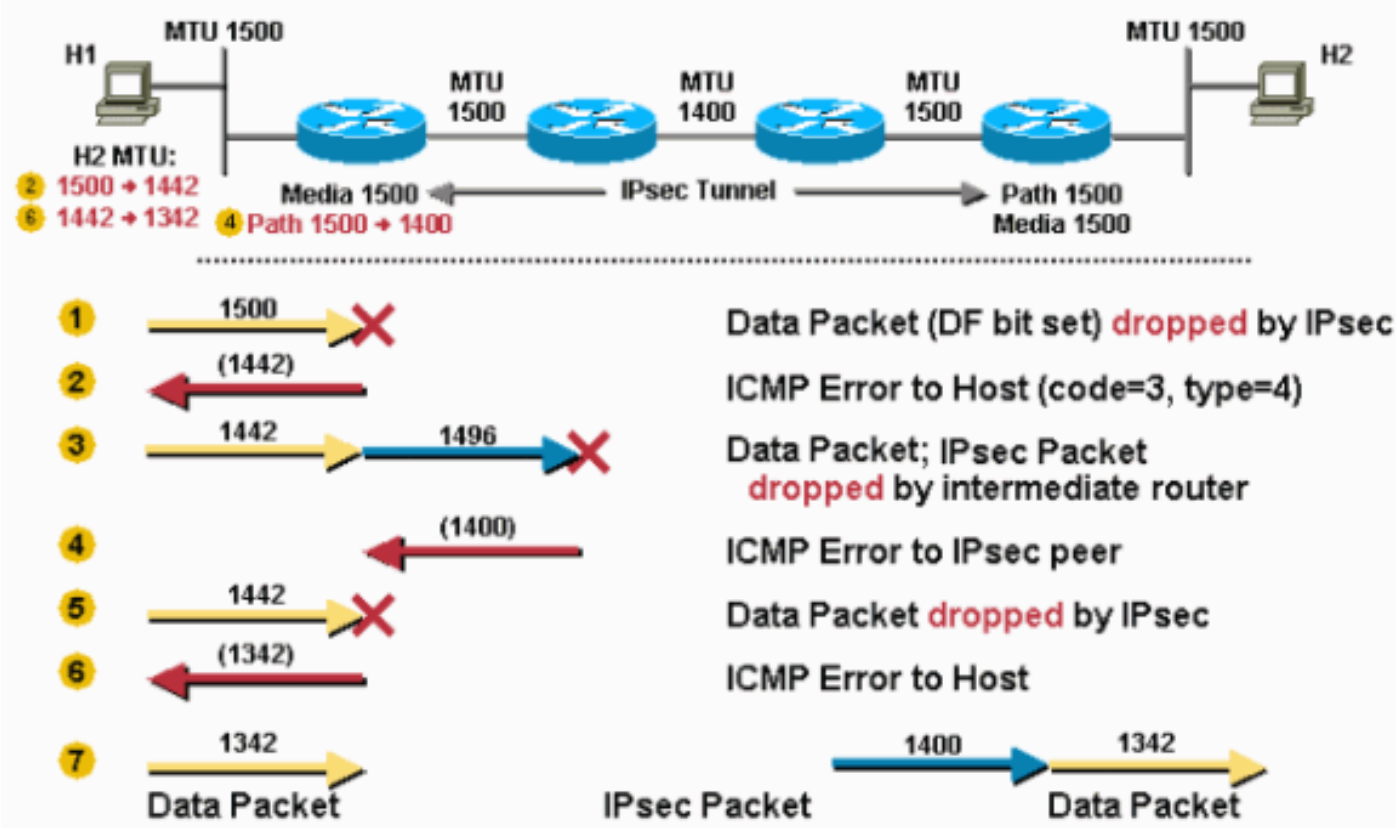
1. Le routeur reçoit un paquet de 1500 octets (en-tête IP de 20 octets + charge utile TCP de 1480 octets) destiné à l'hôte 2.
2. Le paquet de 1500 octets est crypté par IPsec et 52 octets de temps système sont ajoutés (en-tête d'IPsec, queue de bande, et en-tête IP supplémentaire). Maintenant IPsec doit envoyer un paquet de 1552 octets. Puisque le MTU sortant est de 1500, ce paquet devra être fragmenté.
3. Deux fragments sont créés hors du paquet IPsec. Pendant la fragmentation, un en-tête IP de 20 octets supplémentaire est ajouté pour le deuxième fragment, ce qui aboutit à un fragment de 1500 octets et un fragment IP de 72 fragments.
4. Le routeur homologue de tunnel IPsec reçoit les fragments, décolle l'en-tête IP supplémentaire et fusionne les fragments IP dans le paquet IPsec initial. Ensuite, IPsec déchiffre ce paquet.
5. Le routeur achemine ensuite le paquet de données de 1500 octets vers l'hôte 2.

Scénario 8

Ce scénario est semblable au scénario 6, sauf que dans ce cas, le bit DF est défini dans le paquet de données d'origine ; de plus, une liaison existe sur le chemin entre les homologues de tunnel

IPsec portant un MTU plus faible que celui des autres liaisons. Ce scénario explique comment le routeur partenaire d'IPsec exécute les deux rôles PMTUD, comme décrit dans la section Le routeur en tant que participant PMTUD au point d'extrémité d'un tunnel .

Vous verrez dans ce scénario comment PMTU IPsec prend une valeur plus faible en raison du besoin de fragmentation. Rappelez-vous que le bit DF est copié de l'en-tête IP intérieur vers l'en-tête IP extérieur, lorsqu'IPsec crypte un paquet. Les valeurs MTU et PMTU des supports sont stockées dans l'association de sécurité IPsec (SA). Le MTU du support prend comme base le MTU de l'interface du routeur sortant, tandis que le PMTU prend comme base la valeur MTU minimale constatée sur le chemin reliant les homologues IPsec. Rappelez-vous qu'IPsec encapsule/encrypte le paquet avant de tenter de le fragmenter.



1. Le routeur reçoit un paquet de 1500 octets et le supprime car le temps système ISsec, une fois ajouté, rendra le paquet plus volumineux que la valeur de PMTU (1500).
2. Le routeur envoie un message ICMP à l'hôte 1, en indiquant que le MTU du prochain saut est 1442 (1500 - 58 = 1442). Ces 58 octets correspondent au temps système IPsec maximum lors de l'utilisation d'IPsec ESP et d'ESPauth. Le vrai temps système d'IPsec peut être pas moins de 7 octets moins que cette valeur. L'hôte 1 enregistre cette information, habituellement sous forme de route hôte pour la destination (hôte 2), dans sa table de routage.
3. L'hôte 1 ramène son PMTU pour l'hôte 2 à 1442 ; par conséquent, l'hôte 1 envoie des paquets plus petits (1442 octets) lorsqu'il retransmet les données à l'hôte 2. Le routeur reçoit le paquet de 1442 octets et IPsec ajoute 52 octets de temps système de chiffrement ; le paquet IPsec ainsi obtenu est égal à 1496 octets. Puisque le bit DF de ce paquet est défini dans son en-tête, il est supprimé par le routeur du milieu, avec la liaison MTU 1400 octets.
4. Le routeur du milieu qui a supprimé le paquet envoie un message ICMP à l'expéditeur du paquet IPsec (le premier routeur), indiquant que le MTU du prochain saut est égal à 1400 octets. Cette valeur est enregistrée dans le PMTU IPsec SA.
5. La prochaine fois que l'hôte 1 retransmettra un paquet de 1442 octets (il n'a pas reçu

l'accusé de réception correspondant), l'IPsec supprimera le paquet. De nouveau le routeur relâchera le paquet parce que le temps système d'IPsec, quand ajouté au paquet, le rendra plus grand que le PMTU (1400).
6. Le routeur envoie un message ICMP à l'hôte 1, en indiquant que le MTU du prochain saut est désormais 1342. (1400 - 58 = 1342). L'hôte 1 enregistrera de nouveau ces informations.
7. Quand le hôte 1 retransmet de nouveau les données, il utilise le paquet de plus petite taille (1342). Ce paquet n'exigera pas la fragmentation et l'effectuera via le tunnel IPsec à destination de l'hôte 2.

GRE et IPsec ensemble

Des interactions plus complexes pour la fragmentation et le PMTUD se produisent quand IPsec est utilisé afin de chiffrer des tunnels GRE. IPsec et GRE sont combinés de cette manière parce qu'IPsec ne prend en charge pas des paquets de Protocole IP Multicast, ainsi il signifie que vous ne pouvez pas exécuter un protocole de routage dynamique au-dessus du réseau VPN d'IPsec. Les tunnels GRE supportent le multicast ; par conséquent, un tunnel GRE peut être utilisé d'abord pour encapsuler le paquet multicast du protocole de routage dans un paquet unicast IP GRE, qui peut ensuite être crypté par IPsec. Dans ce cas, IPsec est souvent déployé en mode transport en plus de GRE, car les homologues IPsec et les points d'extrémité du tunnel GRE (les routeurs) sont identiques, et le mode transport permettra de gagner 20 octets de temps système IPsec.

Un cas intéressant se présente lorsqu'un paquet IP a été divisé en deux fragments et encapsulé par GRE. Dans ce cas, IPsec verra deux GRE indépendants + paquets IP. Souvent, dans les configurations par défaut, l'un de ces paquets est suffisamment volumineux pour nécessiter une fragmentation une fois le cryptage effectué. L'homologue IPsec devra rassembler ce paquet avant déchiffrement. Cette « double fragmentation » (une fois avant GRE et de nouveau après IPsec) sur le routeur expéditeur augmente la latence et diminue le débit. En outre, le réassemblage est commuté par processus ; par conséquent, on assistera à un hit de CPU sur le routeur récepteur chaque fois que cela se produit.

Cette situation peut être évitée en plaçant « ip mtu » sur l'interface de tunnel GRE assez bas pour prendre en considération le temps système de GRE et d'IPsec (par défaut l'interface de tunnel GRE « ip mtu » est définie sur le MTU de l'interface sortante réelle - octets de temps système GRE).

Ce tableau présente les valeurs suggérées de MTU pour chaque combinaison de tunnel/mode supposant que l'interface physique sortante a un MTU de 1500.

Combinaison de tunnel	MTU spécifique requis	MTU recommandé
GRE + IPsec (transport mode)	1440 bytes	1400 bytes
GRE + IPsec (tunnel mode)	1420 bytes	1400 bytes

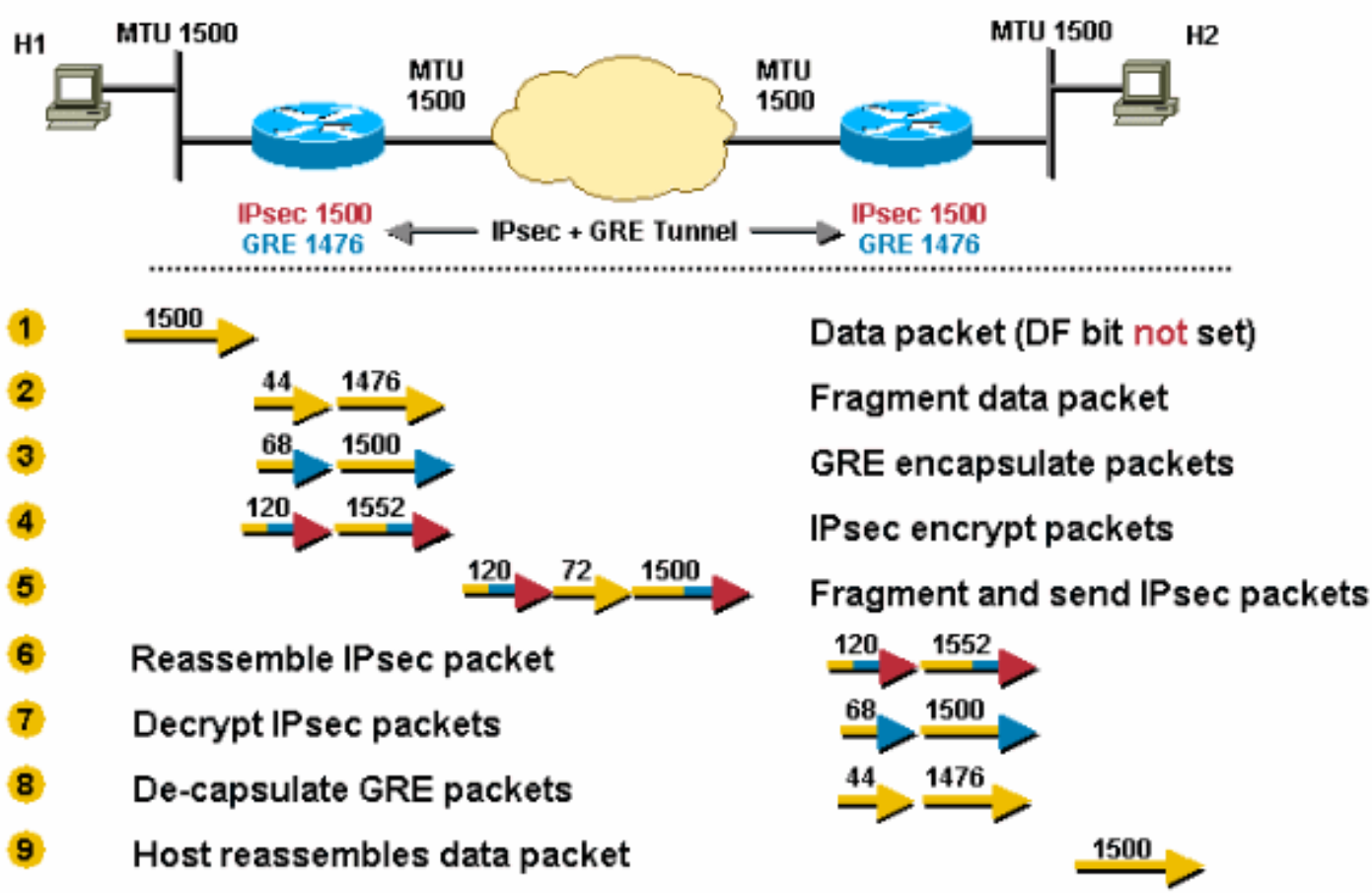
> Remarque: La valeur de MTU de 1400 est recommandée parce qu'elle couvre le plus commun des combinaisons GRE + d'IPsec mode. En outre, il n'y a de du côté incliné pas discernable à tenir compte des 20 ou 40 octets supplémentaires au-dessus. Il est plus facile de se rappeler et de définir une valeur et cette valeur couvre presque tous les scénarios.

Scénario 9

IPsec est déployé sur GRE. Le MTU physique sortant est 1500, l'IPsec PMTU est 1500, et le GRE

est IP MTU 1476 (1500 - 24 = 1476). Pour cette raison, les paquets TCP/IP seront fragmentés deux fois, une fois avant GRE et une fois après IPsec. Le paquet sera réduit en fragments avant GRE encapsulation et un de ces GRE packets sera réduit en fragments de nouveau après IPsec encryption.

Configurant « ip mtu 1440" (IPsec transport mode) ou « ip mtu 1420" (tunnel IPsec mode) sur le tunnel GRE supprime la possibilité de double fragmentation dans ce scénario.



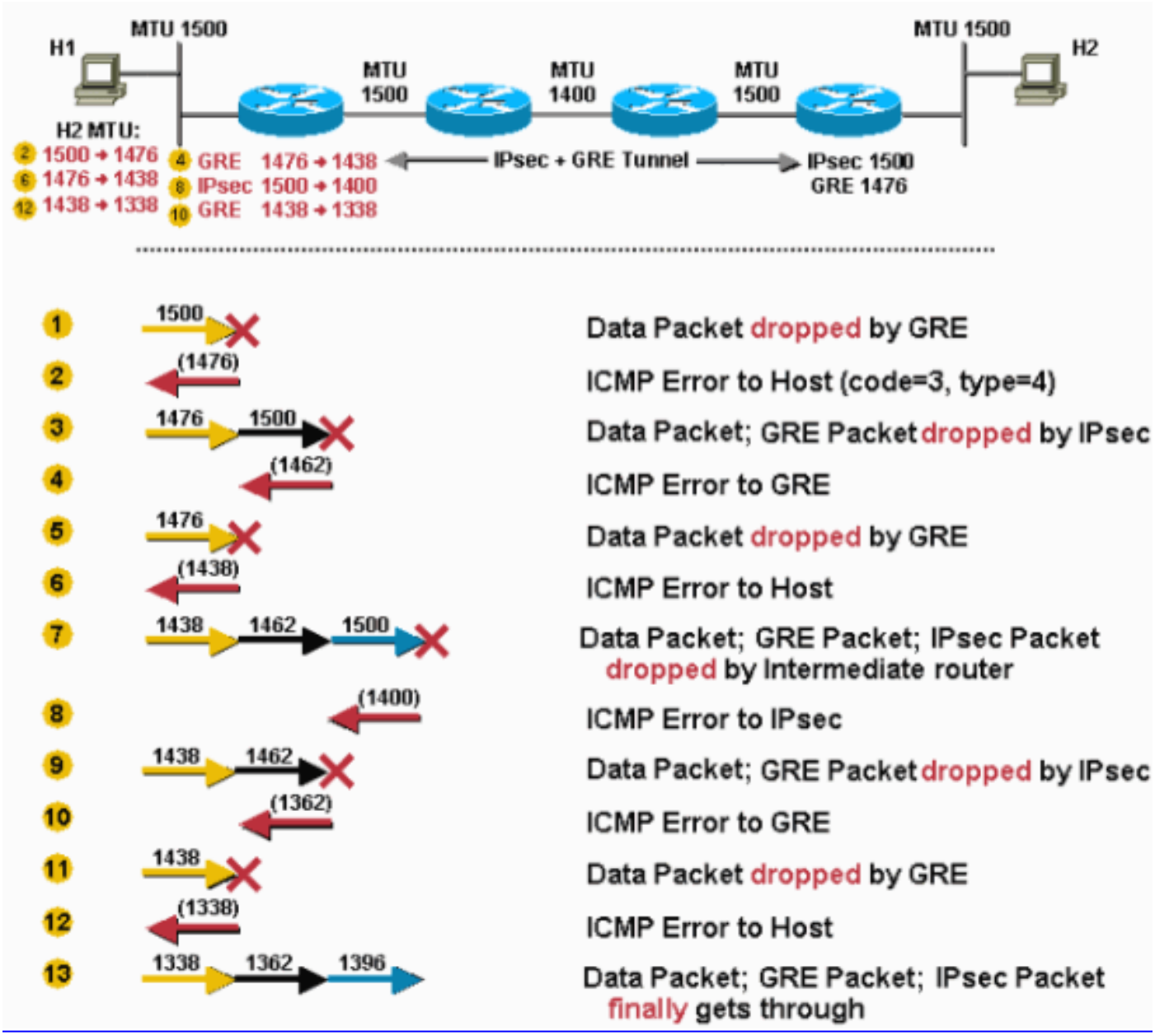
1. Le routeur reçoit un datagramme de 1500 octets.
2. Avant encapsulation, GRE réduit le paquet de 1500 octets dans deux 1476 (1500 - 24 = 1476) et 44 (24 données + 20 en-têtes IP) octets de parties.
3. GRE encapsule les fragments d'IP, qui ajoute 24 octets à chaque paquet. Ceci a comme conséquence deux GRE + paquets IPsecs de 1500 (1476 + 24 = 1500) et 68 (44 + 24) octets chacun.
4. IPsec chiffre les deux paquets, ajoutant 52 byes (tunnel mode d'IPsec) de temps système d'encapsulation à chacun, afin de donner un 1552-byte et un paquet 120-byte.
5. Le paquet 1552-byte IPsec est fragmenté par le routeur parce qu'il est plus grand que le MTU sortant (1500). Le paquet de 1552 octets est divisé en plusieurs parties, un paquet de 1500 octets et un paquet de 72 octets (52 octets de « charge utile » plus une en-tête IP de 20 octets supplémentaires pour le deuxième fragment). Les trois paquets de 1500 octets, 72 octets et 120 octets sont expédiés à IPsec + à l'homologue GRE.
6. Le routeur récepteur rassemble les deux fragments d'IPsec (1500 octets et 72 octets) afin d'obtenir l'original 1552-byte IPsec + le paquet GRE. Rien ne doit être fait au paquet de 120 octets IPsec + GRE.
7. IPsec déchiffre 1552-byte et 120-byte IPsec + paquets GRE afin d'obtenir des paquets 1500-byte et 68-byte GRE.

8. GRE désencapsule les paquets 1500-byte et 68-byte GRE afin d'obtenir le paquet IP 1476-byte et 44-byte fragmente. Ces fragments de paquet IP sont expédiés à l'hôte de destination.
9. L'hôte 2 rassemble ces fragments IP afin d'obtenir le datagramme IP de l'original 1500-byte.

Le scénario 10 est semblable au scénario 8, à l'exception qu'on note une liaison MTU inférieure dans le chemin de tunnel. C'est un scénario du « pire des cas » pour le premier paquet envoyé de l'hôte 1 à l'hôte 2. Après que la dernière étape dans ce scénario, l'hôte 1 place le PMTU correct pour l'hôte 2 et tout est bien pour les connexions TCP entre l'hôte 1 et les écoulements de TCP de l'hôte 2. entre l'hôte 1 et autre héberge (accessible par l'intermédiaire du tunnel d'IPsec + GRE) devront seulement passer par les trois dernières étapes du scénario 10.

Dans ce scénario, la commande de **tunnel path-mtu-discovery** est configurée sur le tunnel GRE et le bit DF est placé sur les paquets TCP/IP qui proviennent de l'hôte 1.

Scénario 10



1. Le routeur reçoit un paquet de 1500 octets. Ce paquet est supprimé par GRE, parce que GRE ne peut pas fragmenter ou expédier le paquet car le bit DF est défini et la taille du paquet dépasse l'interface de sortie « ip mtu » après l'ajout du temps système GRE (24

octets).
2. Le routeur envoie un message ICMP pour héberger 1 afin de le faire savoir que le MTU de prochain-saut est 1476 (1500 - 24 = 1476).
3. L'hôte 1 modifie son PMTU pour l'hôte 2 en 1476 et envoie la plus petite taille lorsqu'il retransmet le paquet. GRE l'encapsule et remet le paquet de 1500 octets à IPsec. IPsec supprime le paquet parce que GRE a copié le bit DF (défini) à partir de l'en-tête IP interne, et avec le temps système IPsec (maximum 38 octets), le paquet est trop grand pour expédier l'interface physique.
4. IPsec envoie un message ICMP à GRE qui indique que le MTU de prochain-saut est de 1462 octets (puisqu'un maximum 38 octets sera ajouté au-dessus pour le cryptage et l'IP). GRE enregistre la valeur 1438 (1462 - 24) en tant que « ip mtu » sur l'interface de tunnel.Remarque: Cette variation de la valeur est enregistrée en interne et ne peut pas être vue dans la sortie de la commande **tunnel<#> de show ip interface** . Vous verrez seulement cette modification si vous utilisez la commande **debug tunnel** .
5. La prochaine fois que l'hôte 1 retransmettra le paquet de 1476 octets, GRE le supprimera.
6. Le routeur envoie un message ICMP pour héberger 1 qui indique que 1438 est le MTU de prochain-saut.
7. L'hôte 1 diminue le PMTU pour l'hôte 2 et retransmet un paquet de 1438 octets. Cette fois, GRE accepte le paquet, l'encapsule, et le remet à IPsec pour cryptage. Le paquet IPsec est expédié au routeur intermédiaire et supprimé parce qu'il a un MTU d'interface de sortie de 1400.
8. Le routeur intermédiaire envoie un message ICMP à IPsec qui lui indique que le MTU de prochain-saut est 1400. Cette valeur est enregistrée par IPsec dans la valeur PTMU du SA IPsec associé.
9. Quand l'hôte 1 retransmet le paquet de 1438 octets, GRE l'encapsule et le remet à IPsec. IPsec supprime le paquet parce qu'il a changé son propre PMTU en 1400.
10. IPsec envoie une erreur d'ICMP à GRE qui indique que le MTU de prochain-saut est 1362, et GRE enregistre la valeur 1338 intérieurement.
11. Quand l'hôte 1 retransmet le paquet d'origine (car il n'a pas reçu d'accusé de réception), GRE le supprime.
12. Le routeur envoie un message ICMP pour héberger 1 qui indique que le MTU de prochain-saut est 1338 (1362 - 24 octets). L'hôte 1 abaisse son PMTU pour l'hôte 2 à 1338.
13. L'hôte 1 retransmet un paquet de 1338 octets, et cette fois, il peut finalement accéder complètement à l'hôte 2.

Autres recommandations

La configuration de la commande **tunnel path-mtu-discovery** sur une interface de tunnel peut favoriser l'interaction entre GRE et IPsec quand elles sont configurées sur le même routeur. Rappelez-vous que si la commande **tunnel path-mtu-discovery** n'est pas configurée, le bit DF est systématiquement effacé dans l'en-tête IP GRE. Ceci permet le paquet IP GRE à fragmenter quoique l'en-tête IP encapsulée de données ait eu le bit DF réglé, qui normalement ne permettrait pas le paquet à fragmenter.

Si la commande de **tunnel path-mtu-discovery** est configurée sur l'interface de tunnel GRE, ceci se produira.

1. GRE copiera le bit DF de l'en-tête IP vers l'en-tête IP GRE.

2. Si le bit DF est défini dans l'en-tête IP GRE et que le paquet est trop volumineux après le cryptage IPsec pour le MTU IP de l'interface physique de sortie, IPsec supprime le paquet et notifie le tunnel GRE en vue d'une réduction de sa taille MTU IP.
3. IPsec fait PMTUD pour ses propres paquets et si l'IPsec PMTU change (s'il est réduit), alors IPsec n'informe pas immédiatement GRE, mais quand un autre paquet « trop grand » est livré complet, alors le processus dans l'étape 2 se produit.
4. Le MTU IP de GRE est maintenant plus petit ; par conséquent, il supprimera n'importe quel paquet de données IP avec le bit DF défini (devenu trop grand) et enverra un message ICMP à l'hôte expéditeur.

La commande **tunnel path-mtu-discovery** aide commande l'interface GRE à définir dynamiquement son MTU IP, plutôt que de le faire de façon statique avec la commande **ip mtu** . Il est recommandé d'utiliser ces deux commandes. La commande **ip mtu** est utilisée pour fournir au GRE l'espace et le temps système d'IPsec relatifs au MTU IP de l'interface physique sortante locale. La commande **tunnel path-mtu-discovery** permet de diminuer davantage le MTU IP du tunnel GRE s'il existe une liaison MTU IP dans le chemin entre les homologues IPsec.

Voici ce que vous pouvez faire si vous rencontrez des problèmes PMTUD au sein d'un réseau sur lequel GRE + des tunnels IPsec sont configurés.

Cette liste commence par la solution la plus desirable.

- Réparez le problème avec PMTUD ne fonctionnant pas, qui est habituellement provoqué par un routeur ou un Pare-feu qui bloque l'ICMP.
- Utilisez la commande **ip tcp adjust-mss** sur les interfaces de tunnel afin que le routeur diminue la valeur MSS TCP au niveau du paquet TCP SYN. Cela aide les deux hôtes d'extrémité (l'expéditeur et le destinataire TCP) à utiliser des paquets suffisamment petits pour ne pas nécessiter l'intervention de PMTUD.
- Utilisez le routage spécifique sur l'interface d'entrée du routeur et configurez une route pour effacer le bit DF de l'en-tête IP avant l'accès à l'interface de tunnel GRE. Ceci permettra la fragmentation du paquet de données IP avant l'encapsulation GRE.
- Augmentez la valeur « ip mtu » sur l'interface de tunnel GRE pour qu'elle soit égale à la valeur MTU de l'interface de sortie. Ceci permettra au paquet de données IP d'être encapsulé sans fragmentation préalable. Le paquet GRE sera alors crypté par IPsec, puis fragmenté pour la sortie de l'interface physique de sortie. Dans ce cas, vous ne configurez pas la commande **tunnel path-mtu-discovery** sur l'interface de tunnel GRE. Ceci peut diminuer considérablement le débit, car le réassemblage de paquets IP sur l'homologue IPsec s'effectue en mode de commutation de processus.

Informations connexes

- Page de support pour le routage IP
- Page d'assistance d'IPSec (protocole de sécurité IP)
- Calculatrice supplémentaire d'IPSec (calculez la longueur de paquet avec des protocoles d'encapsulation d'IPSec)
- Découverte de MTU RFC 1191
- Options de découverte RFC 1063 IP MTU
- Internet Protocol RFC 791
- RFC 793 Transmission Control Protocol (TCP)

- RFC 879 Taille maximale des segments TCP et rubriques connexes
- RFC 1701 Encapsulation de routage générique (GRE)
- RFC 1241 Plan pour protocole IEP (Internet Encapsulation Protocol)
- Encapsulation RFC 2003 IP dans IP
- Support technique - Cisco Systems