Risoluzione dei problemi di IPv4 Fragmentation, MTU, MSS e PMTUD con GRE e IPsec

Sommario

- Introduzione
- Premesse
- Frammentazione dei pacchetti IPv4 e riassemblaggio
  - Problemi di frammentazione IPv4
  - Come evitare la frammentazione IPv4: funzionamento del parametro TCP MSS
    - Esempio 1
    - Esempio 2
- Informazioni sulla funzionalità PMTUD
    - Esempio 3
    - Esempio 4
  - Problemi della funzionalità PMTUD
  - Topologie di rete comuni che richiedono l'uso del PMTUD
- Tunnel
  - Considerazioni sulle interfacce tunnel
  - Ruoli svolti dal router durante il processo PMTUD sull'endpoint del tunnel
    - Esempio 5
    - Esempio 6
- Modalità tunnel IPsec puro
    - Esempio 7
    - Esempio 8
- Uso congiunto di GRE e IPv4sec
    - Esempio 9
    - Esempio 10
- Ulteriori suggerimenti
- Informazioni correlate

Introduzione

In questo documento viene descritto come funzionano la frammentazione IPv4 e il rilevamento PMTUD (Path Maximum Transmission Unit Discovery).

Premesse

Vengono inoltre discussi gli scenari in cui viene descritto il comportamento della funzionalità PMTUD rispetto a diverse configurazioni di tunnel IPv4.

Frammentazione dei pacchetti IPv4 e riassemblaggio

Anche se la lunghezza massima di un datagramma IPv4 è 65535 byte, la maggior parte dei collegamenti di trasmissione applica un limite inferiore, con il valore MTU. Il valore MTU dipende dal collegamento di trasmissione.

Il protocollo IPv4 supporta MTU di diversa lunghezza e consente ai router di frammentare i datagrammi IPv4 secondo necessità.

La stazione ricevente è responsabile del riassemblaggio dei frammenti nel datagramma IPv4 alle dimensioni originarie.

La frammentazione IPv4 suddivide un datagramma in parti che vengono ricomposte successivamente.

Per la frammentazione e il riassemblaggio dell'IPv4, vengono usati i campi origine, destinazione, identificazione, lunghezza totale e posizione relativa del frammento, oltre ai flag "more fragments" (MF) e "do not fragment" (DF) nell'intestazione.

Per ulteriori informazioni sulla frammentazione di pacchetti IPv4 e il riassemblaggio, consultare la RFC 791.

Questa immagine mostra il layout di un'intestazione 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

Il campo Identification (identificazione) è lungo 16 bit ed è un valore assegnato dal mittente di un datagramma IPv4. Ciò semplifica il riassemblaggio dei frammenti di un datagramma.

Il campo Fragment Offset (posizione relativa del frammento) è lungo 13 bit e permette di individuare la posizione del frammento nel datagramma IPv4 originale. Questo valore è un multiplo di 8 byte.

Il campo Flags dell'intestazione IPv4 contiene 3 bit per i flag di controllo. Il bit "non frammentare" (DF, Don't Fragment) determina se un pacchetto può essere frammentato o meno.

Il bit 0 è riservato ed è sempre impostato su 0.

Il bit 1 è il bit DF (0 = "puoi frammentare", 1 = "non frammentare").

Il bit 2 è il bit MF (0 = "ultimo frammento", 1 = "altri frammenti").

Valore	Bit 0 riservato	Bit 1 DF	Bit 2 MF
0	0	Puoi frammentare	Ultimo
1	0	Non frammentare	Altri

Aggiungendo le lunghezze dei frammenti IPv4, il valore supera il datagramma IPv4 originale di 60 byte.

La lunghezza complessiva è maggiore di 60 in quanto sono state create tre ulteriori intestazioni IPv4, una per ciascun frammento successivo al primo.

La posizione relativa del primo frammento è 0, la sua lunghezza 1500, inclusi i 20 byte dell'intestazione IPv4 originale leggermente modificata.

La posizione relativa del secondo frammento è 185 (185 x 8 = 1480); la parte dati di questo frammento inizia al byte 1480 del datagramma IPv4 originale.

Il frammento è lungo 1500 byte, inclusa l'intestazione IPv4 aggiuntiva creata appositamente.

La posizione relativa del terzo frammento è 370 (370 x 8 = 2960), quindi la parte dati di questo frammento inizia al byte 2960 del datagramma IPv4 originale.

Il frammento è lungo 1500 byte, inclusa l'intestazione IPv4 aggiuntiva creata appositamente.

La posizione relativa del quarto frammento è 555 (555 x 8 = 4440), quindi la parte dati di questo frammento inizia al byte 4440 del datagramma IPv4 originale.

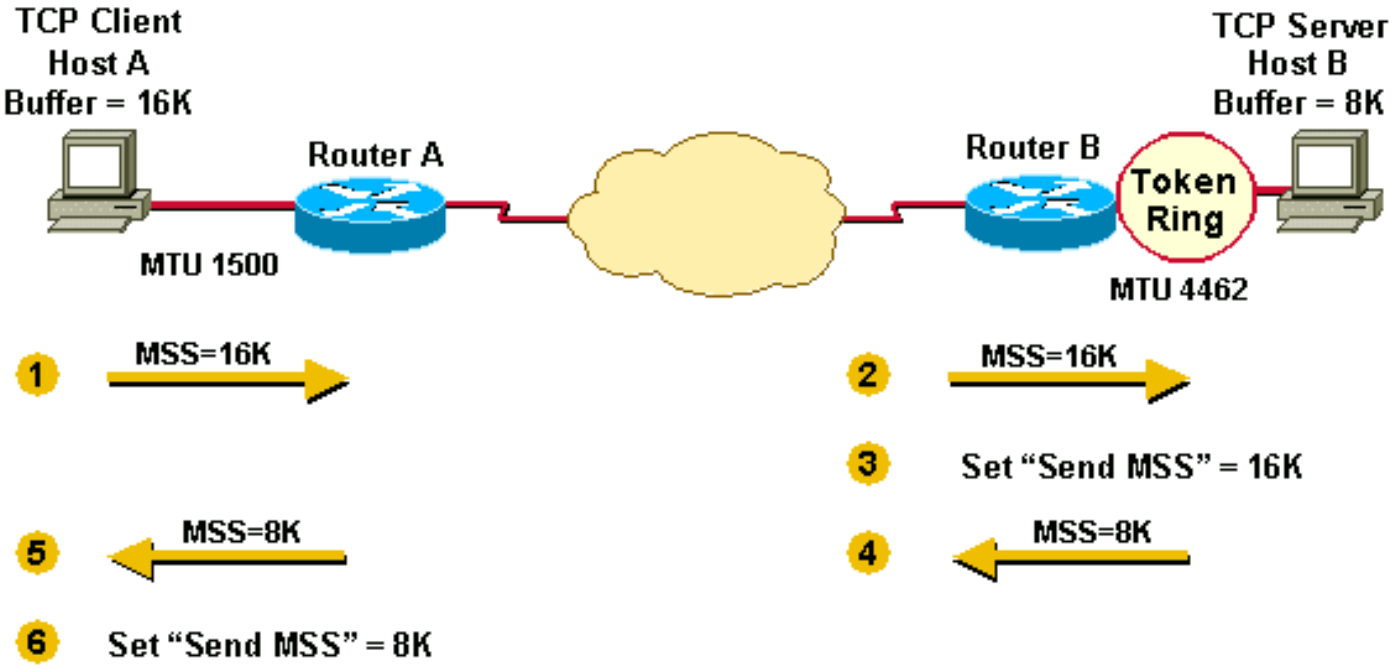
Questo frammento è lungo 700 byte, inclusa l'intestazione IPv4 aggiuntiva creata appositamente.

Le dimensioni del datagramma IPv4 originale possono essere determinate solo dopo aver ricevuto l'ultimo frammento.

La posizione relativa dell'ultimo frammento (555) equivale a 4440 byte nel datagramma IPv4 originale.

La somma dei byte di dati dell'ultimo frammento (680 = 700 - 20) restituisce 5120 byte, ossia la parte dati del datagramma IPv4 originale.

L'aggiunta dei 20 byte di un'intestazione IPv4 equivale alle dimensioni del datagramma IPv4 originale (4440 + 680 + 20 = 5140), come mostrato nelle immagini.



Problemi di frammentazione IPv4

La frammentazione IPv4 provoca un lieve aumento del sovraccarico della CPU e della memoria per frammentare un datagramma IPv4. Ciò vale sia per il mittente sia per i router eventualmente posizionati tra il mittente e il destinatario.

La creazione di frammenti implica la creazione di intestazioni e la copia del datagramma originale nei frammenti.

Questa operazione viene effettuata in modo efficiente in quanto le informazioni necessarie per creare i frammenti sono subito disponibili.

La frammentazione provoca un maggiore sovraccarico per il destinatario nella fase di riassemblaggio, in quanto il destinatario deve riservare parte della memoria ai frammenti in arrivo e, una volta ricevuti tutti, ricomporli nel datagramma.

Il riassemblaggio su un host non è considerato un problema perché l'host dispone del tempo e delle risorse di memoria da dedicare a questa attività.

Al contrario, sul router, il cui compito principale è inoltrare i pacchetti nel più breve tempo possibile, l'operazione di riassemblaggio non è efficiente.

Il modo con cui è progettato un router non gli permette di tenere i pacchetti per periodi prolungati.

Per il riassemblaggio il router deve allocare il buffer più grande a sua disposizione (18K) perché non ha modo di determinare le dimensioni del pacchetto IPv4 originale finché non riceve l'ultimo frammento.

La gestione dei frammenti scartati è un altro problema della frammentazione.

Se si elimina un frammento di un datagramma IPv4, sarà necessario che sia presente l'intero

datagramma IPv4 originale, che verrà a sua volta frammentato.

Questa condizione viene rilevata con il Network File System (NFS). Il NFS ha blocchi in lettura e in scrittura di 8192 byte.

Pertanto, un datagramma NFS IPv4/UDP è lungo circa 8500 byte (incluse le intestazioni NFS, UDP e IPv4).

Una stazione di invio collegata a Ethernet (MTU 1500) deve frammentare il datagramma da 8500 byte in sei (6) frammenti, di cui cinque (5) da 1500 byte e uno (1) da 1100 byte.

Se uno dei sei frammenti viene scartato a causa di traffico eccessivo sul collegamento, sarà necessario ritrasmettere il datagramma originale completo. Il risultato sono altri sei frammenti da creare.

Se questo collegamento scarta un pacchetto su sei, le probabilità di trasferimento dei dati NFS su questo collegamento sono basse, in quanto almeno un frammento IPv4 verrebbe scartato da ciascun datagramma NFS IPv4 originale da 8500 byte.

I firewall che filtrano o gestiscono i pacchetti in base alle informazioni dal layer 4 (L4) al layer 7 (L7) hanno problemi nell'elaborare correttamente i frammenti IPv4.

Se i frammenti IPv4 non sono ordinati, un firewall blocca i frammenti non iniziali perché non contengono le informazioni corrispondenti al filtro del pacchetto.

Il datagramma IPv4 originale potrebbe non essere riassemblato dall'host ricevente.

Se il firewall è configurato in modo da filtrare anche i frammenti non iniziali che non hanno informazioni sufficienti, potrebbe crearsi una vulnerabilità nel sistema.

I dispositivi di rete, ad esempio i Content Switch Engine, indirizzano i pacchetti in base alle informazioni da L4 a L7 e, se un pacchetto è stato suddiviso in più frammenti, il dispositivo non riesce a rispettarne le policy.

Come evitare la frammentazione IPv4: funzionamento del parametro TCP MSS

Il parametro Maximum Segment Size (MSS) del segmento TCP (Transmission Control Protocol) definisce la quantità massima di dati accettata da un host in un datagramma TCP/IPv4.

Questo datagramma TCP/IPv4 potrebbe essere frammentato sul layer IPv4. Il valore MSS viene inviato come opzione dell'intestazione TCP solo nei segmenti SYN di TCP.

Ogni lato di una connessione TCP segnala il proprio valore MSS al lato opposto. Il valore MSS non è oggetto di negoziazione tra gli host.

L'host mittente deve limitare le dimensioni dei dati in un singolo segmento TCP a un valore inferiore o uguale al valore MSS segnalato dall'host ricevente.

In origine, il valore MSS indicava la quantità di buffer (pari ad almeno 65496 byte) riservata su una stazione ricevente per memorizzare i dati TCP contenuti in un datagramma IPv4.

Il valore MSS era il segmento di dati delle dimensioni massime che il ricevitore TCP era disposto ad accettare. Questo segmento TCP poteva essere grande 64K e frammentato sul layer IPv4 per essere trasmesso all'host ricevente.

L'host ricevente avrebbe quindi ricomposto il datagramma IPv4 prima di consegnare il segmento TCP completo al layer TCP.

Come vengono impostati e usati i valori MSS per limitare le dimensioni del segmento TCP e del datagramma IPv4.

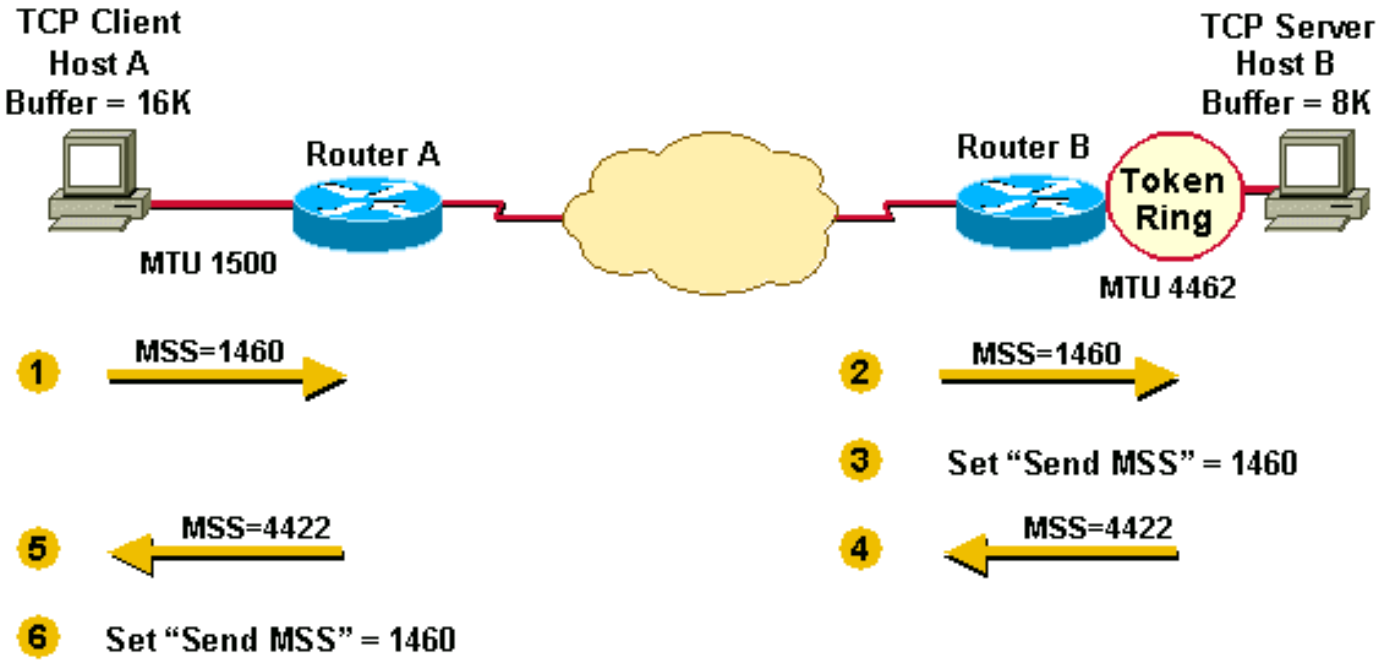
Nell'esempio 1 viene mostrato come il valore MSS è stato implementato all'inizio.

L'host A ha un buffer di 16K, l'host B un buffer di 8K. I due host inviano e ricevono i rispettivi valori MSS e concordano sul valore MSS di invio per poter comunicare tra loro.

L'host A e l'host B devono frammentare i datagrammi IPv4 che sono più grandi della MTU dell'interfaccia, ma più piccoli del valore MSS di invio, in quanto lo stack TCP passa 16K o 8K byte di dati a IPv4.

Nel caso dell'host B, i pacchetti vengono frammentati per accedere alla LAN Token Ring e quindi di nuovo per accedere alla LAN Ethernet.

Esempio 1



1. L'host A invia il valore MSS di 16K all'host B.
2. L'host B riceve il valore MSS di 16K dall'host A.
3. L'host B imposta il valore MSS di invio a 16K.
4. L'host B invia il valore MSS di 8K all'host A.
5. L'host A riceve il valore MSS di 8K dall'host B.
6. L'host A imposta il valore MSS di invio a 8K.

Per evitare la frammentazione IPv4 sugli endpoint della connessione TCP, è stato selezionato un valore MSS pari alle dimensioni minime del buffer e alla MTU dell'interfaccia in uscita (-40).

I valori MSS sono inferiori ai valori MTU di 40 byte in quanto il valore MSS (ossia le dimensioni dei dati TCP) non include l'intestazione IPv4 da 20 byte e l'intestazione TCP da 20 byte.

Il valore MSS si basa sulle dimensioni predefinite dell'intestazione; sullo stack mittente occorre sottrarre i byte dell'intestazione IPv4 e l'intestazione TCP dipende dalle opzioni TCP o IPv4 usate.

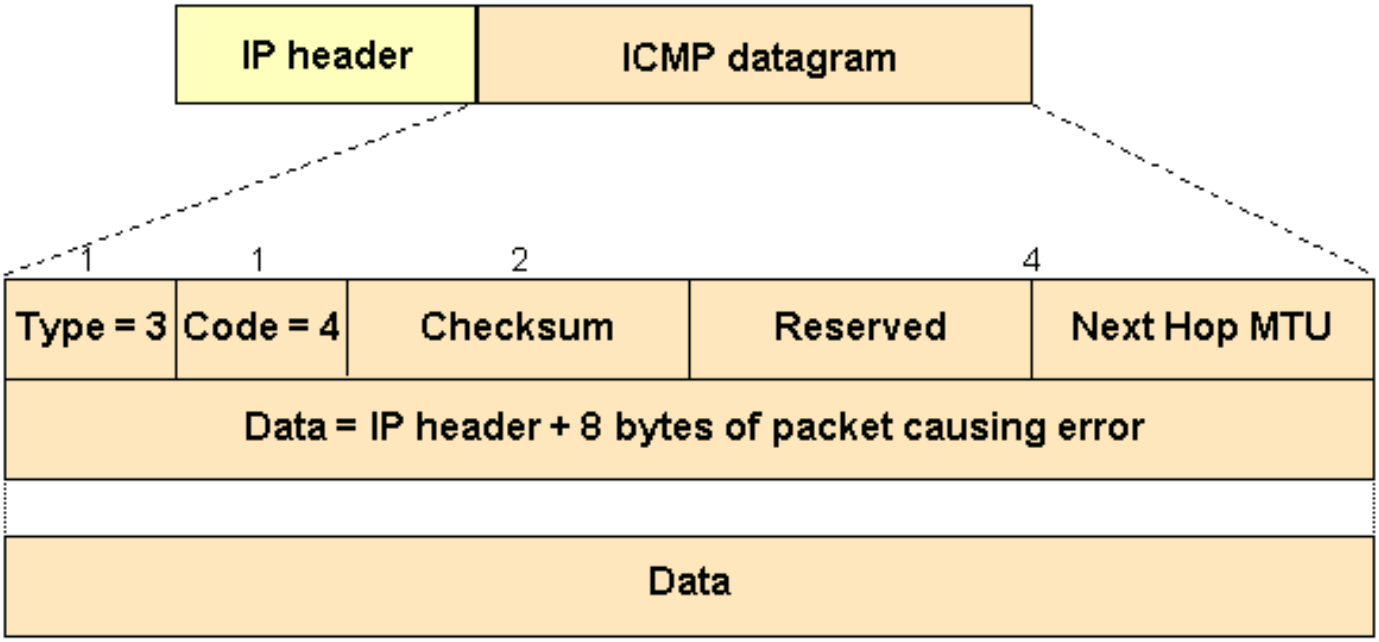
Il valore MSS attualmente funziona in modo tale che ciascun host confronta per primo l'MTU dell'interfaccia in uscita con le dimensioni del proprio buffer e sceglie il valore più basso come valore MSS di invio.

Gli host confrontano quindi le dimensioni MSS ricevute con la propria MTU dell'interfaccia e scelgono nuovamente il valore più basso.

Nell'esempio 2 viene illustrata l'ulteriore azione intrapresa dall'host mittente per evitare la frammentazione sui collegamenti locale e remoto.

Ciascun host prende in considerazione l'MTU dell'interfaccia in uscita prima di inviare i rispettivi valori MSS. Ciò contribuisce a evitare la frammentazione.

Esempio 2



1. L'host A confronta il proprio buffer MSS (16K) e la relativa MTU (1500 - 40 = 1460) e usa il valore più basso (1460) come valore MSS da inviare all'host B.
2. L'host B riceve il valore MSS (1460) inviato dall'host A e lo confronta con il valore della MTU dell'interfaccia in uscita, diminuito di 40 (4422).
3. L'host B sceglie il valore più basso (1460) come valore MSS per inviare i datagrammi IPv4 all'host A.
4. L'host B confronta il proprio buffer MSS (8K) e la relativa MTU (4462 - 40 = 4422) e usa il

valore 4422 come valore MSS da inviare all'host A.
5. L'host A riceve il valore MSS (4422) inviato dall'host B e lo confronta con il valore della MTU dell'interfaccia in uscita, diminuito di 40 (1460).
6. L'host A sceglie il valore più basso (1460) come valore MSS per inviare i datagrammi IPv4 all'host B.

Il valore scelto da entrambi gli host come valore MSS di invio è 1460. Spesso il valore MSS di invio coincide su entrambe le estremità della connessione TCP.

Nell'esempio 2, la frammentazione sugli endpoint della connessione TCP non viene effettuata perché gli host prendono in considerazione le MTU di entrambe le interfacce in uscita.

I pacchetti vengono ancora frammentati sulla rete tra il router A e il router B, in caso un collegamento abbia una MTU inferiore a quella dell'interfaccia in uscita di entrambi gli host.

Informazioni sulla funzionalità PMTUD

Il parametro TCP MSS gestisce la frammentazione sui due endpoint di una connessione TCP, ma non interviene in caso tra i due endpoint vi sia un collegamento con MTU inferiore.

La funzionalità PMTUD è stata sviluppata per evitare la frammentazione nel percorso tra gli endpoint. Viene usata per determinare in modo dinamico il valore MTU più basso nel percorso tra un'origine pacchetto e la sua destinazione.

Nota: la funzionalità PMTUD è supportata solo sui protocolli TCP e UDP. Non è possibile usarla con altri protocolli. se la funzionalità PMTUD è abilitata su un host, il bit DF è impostato su tutti i pacchetti TCP e UDP dell'host.

Quando un host invia un pacchetto di dati MSS completo con bit DF impostato e il pacchetto deve essere frammentato, il PMTUD riduce il valore MSS di invio per la connessione.

Un host registra il valore MTU di una destinazione, perché nella tabella di routing crea una voce host (/32) con questo valore MTU.

Se un router tenta di inoltrare un datagramma IPv4 (con bit DF impostato) su un collegamento la cui MTU è inferiore alle dimensioni del pacchetto, il router scarta il pacchetto e restituisce un messaggio ICMP (Internet Control Message Protocol) "Destination Unreachable" (Destinazione irraggiungibile) all'origine del datagramma IPv4 con il codice "fragmentation needed and DF set" (frammentazione richiesta e DF impostato) (tipo 3, codice 4).

Quando la stazione di origine riceve il messaggio ICMP, diminuisce il valore MSS di invio e, quando il protocollo TCP trasmette nuovamente il segmento, usa le dimensioni del segmento più piccole.

Di seguito è riportato un esempio di messaggio ICMP "frammentazione richiesta e DF impostato" visualizzato su un router dopo l'attivazione del **debug ip icmp** comando:

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

Il diagramma mostra il formato dell'intestazione ICMP di un messaggio "frammentazione richiesta e DF impostato" "destinazione irraggiungibile".

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

In base alla RFC 1191, un router che restituisce un messaggio ICMP del tipo "frammentazione richiesta e DF impostato" deve includere l'MTU della rete dell'hop successivo nei 16 bit meno

significativi del campo dell'intestazione aggiuntiva ICMP contrassegnata come "unused" (inutilizzata) nella RFC 792.

Nelle prime implementazioni della RFC 1191, le informazioni MTU dell'hop successivo non venivano fornite. Anche quando tali informazioni erano fornite, alcuni host le ignoravano.

In questo caso, la RFC 1191 contiene anche una tabella in cui sono elencati i valori suggeriti con cui diminuire l'MTU durante il rilevamento dell'MTU del percorso (PMTUD).

Viene usato dagli host per ottenere più rapidamente un valore ragionevole per il valore MSS di invio, come mostrato nell'esempio.

Il rilevamento dell'MTU del percorso viene eseguito continuamente su tutti i pacchetti perché il percorso tra mittente e destinatario può variare in modo dinamico.

Ogni volta che un mittente riceve messaggi ICMP "Cannot Fragment" (Impossibile frammentare), aggiorna le informazioni di routing (in cui memorizza il PMTUD).

Durante il rilevamento dell'MTU del percorso (PMTUD), possono verificarsi due situazioni:

1. Il pacchetto può arrivare al destinatario senza essere frammentato.

Nota: per proteggere la CPU dagli attacchi DoS, il router limita a due al secondo il numero di messaggi ICMP "destinazione irraggiungibile" che avrebbe inviato. Pertanto, se in uno scenario di rete in cui ci si aspetta che il router risponda con più di due messaggi ICMP (tipo = 3, codice = 4) al secondo (variabile a seconda dell'host), disattivare la limitazione dei messaggi ICMP, è possibile usare il **no ip icmp rate-limit unreachable [df] interface** comando.

---

2. Il mittente riceve messaggi ICMP "Cannot Fragment" (Impossibile frammentare) da hop sul percorso verso il destinatario.

Il rilevamento dell'MTU del percorso (PMTUD) viene eseguito in modo indipendente su entrambe le direzioni di una connessione TCP.

In alcuni casi, il PMTUD in una direzione di flusso attiva una delle stazioni terminali per ridurre il valore MSS di invio, mentre l'altra stazione terminale mantiene il valore MSS di invio originale perché non ha mai inviato un datagramma IPv4 delle dimensioni sufficienti per attivare il PMTUD.

Un esempio è la connessione HTTP illustrata nell'Esempio 3. Il client TCP invia pacchetti piccoli, mentre il server invia pacchetti di grandi dimensioni.

In questo caso, solo i pacchetti di grandi dimensioni provenienti dal server (superiori a 576 byte) fanno attivare il meccanismo PMTUD.

I pacchetti provenienti dal client sono piccoli (inferiori a 576 byte) e non attivano il PMTUD perché non devono essere frammentati per attraversare il collegamento la cui MTU è 576.

Esempio 3

Nell'esempio 4 viene mostrato un esempio di routing asimmetrico in cui uno dei percorsi ha una MTU minima inferiore all'altro.

Il routing asimmetrico si verifica quando per scambiarsi i dati due endpoint usano percorsi diversi.

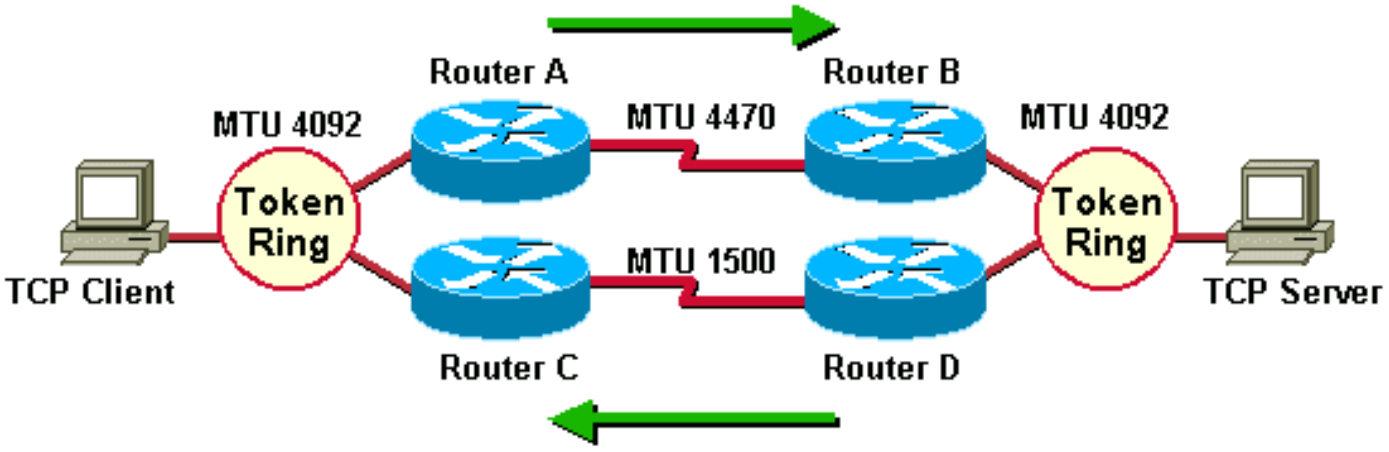
Nell'esempio, il PMTUD attiva la riduzione del valore MSS di invio solo in una direzione della connessione TCP.

Il traffico tra il client TCP e il server passa attraverso il router A e il router B, mentre il traffico di ritorno tra il server e il client passa attraverso il router D e il router C.

Quando il server TCP invia i pacchetti al client, il PMTUD comunica al server di ridurre il valore MSS di invio in quanto, prima di poterli inviare al router C, i pacchetti da 4092 byte devono essere frammentati dal router D.

Al contrario, il client non riceve mai un messaggio ICMP "Destination Unreachable" (Destinazione irraggiungibile) con il codice "fragmentation needed and DF set" (frammentazione richiesta e DF impostato) in quanto il router A non deve frammentare i pacchetti quando li invia al server tramite il router B.

Esempio 4



Nota: il comando ip tcp path-mtu-discovery viene usato per abilitare il rilevamento della MTU del percorso sulle connessioni TCP inizializzate dai router (ad esempio, BGP e Telnet).

Problemi della funzionalità PMTUD

Il processo PMTUD può essere interrotto.

- Un router scarta un pacchetto e non invia un messaggio ICMP. (raro).
- Un router genera e invia un messaggio ICMP ma il messaggio ICMP rimane bloccato da un router o da un firewall tra il router e il mittente (comune).
- Un router genera e invia un messaggio ICMP ma il mittente ignora il messaggio (raro).

Il primo e l'ultimo dei tre punti elenco sono in genere il risultato di un errore, ma il punto centrale descrive un problema comune.

In genere, quando si applicano i filtri ai pacchetti ICMP, si tende a bloccare tutti i tipi di messaggi anziché selezionarli opportunamente.

Il filtro può bloccare tutti i tipi di messaggi ICMP ad eccezione dei messaggi "destinazione irraggiungibile" o "tempo scaduto".

La riuscita del processo PMTUD si basa sui messaggi ICMP "destinazione irraggiungibile" inviati dal mittente di un pacchetto TCP/IPv4.

I messaggi ICMP "time-exceeded" (tempo scaduto) sono rilevanti per altre problematiche del protocollo IPv4.

Di seguito è riportato un esempio di applicazione del filtro al router.

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

Per evitare che il protocollo ICMP venga bloccato del tutto, è possibile usare altre tecniche.

- Annullare il bit DF sul router e permettere la frammentazione. (Non è una buona idea, però. per approfondimenti, vedere la sezione dedicata ai problemi di frammentazione IP).
- Modificare il valore TCP MSS con il comando **ip tcp adjust-mss <500-1460>**interface.

Nell'esempio successivo, il router A e il router B si trovano nello stesso dominio amministrativo. Non è possibile raggiungere il router C e l'ICMP è bloccato, il processo PMTUD viene interrotto.

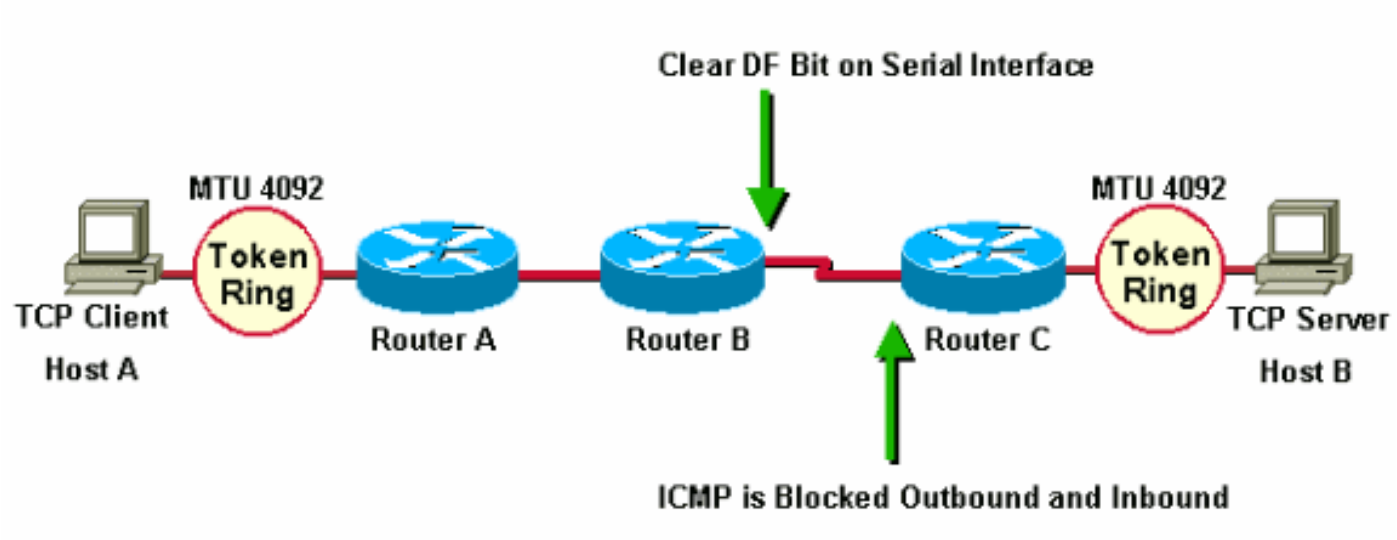
Per risolvere questa situazione, è necessario annullare il bit DF sul router B in entrambe le direzioni e consentire la frammentazione. A tale scopo, è possibile eseguire il routing delle policy.

La sintassi per annullare il bit DF è disponibile a partire da Cisco IOS® versione 12.1(6).

```
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
```



In alternativa è possibile modificare il valore dell'opzione TCP MSS sui pacchetti SYN che attraversano il router (disponibile su Cisco IOS® versione 12.2(4)T e successive).

In questo modo il valore MSS nel pacchetto TCP SYN viene abbassato in modo che sia inferiore al valore (1460) specificato nel **ip tcp adjust-mss**comando.

Di conseguenza, il mittente TCP non invia segmenti più grandi di questo valore.

Il pacchetto IPv4 è più grande del valore MSS (1460 byte) di 40 byte (1500) in modo da tenere conto anche dell'intestazione TCP (20 byte) e dell'intestazione IPv4 (20 byte).

È possibile cambiare il valore MSS dei pacchetti TCP SYN con il **ip tcp adjust-mss** comando. Questa sintassi riduce a 1460 il valore MSS sui segmenti TCP.

Il comando permette di instradare il traffico in entrata e in uscita sull'interfaccia serial0.

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

Con l'aumento dei tunnel IPv4, sono aumentati anche i problemi di frammentazione IPv4.

I tunnel causano una maggiore frammentazione perché l'incapsulamento del tunnel aggiunge un "carico ulteriore" alle dimensioni di un pacchetto.

Ad esempio, l'uso del GRE (Generic Router Encapsulation) aggiunge 24 byte a un pacchetto e,

dopo questo aumento, il pacchetto deve essere frammentato perché è più grande dell'MTU in uscita.

Topologie di rete comuni che richiedono l'uso del PMTUD

Il PMTUD è richiesto quando i collegamenti intermedi hanno MTU inferiori rispetto alle MTU dei collegamenti terminali. Alcuni dei motivi più comuni dell'esistenza di questi collegamenti MTU più piccoli sono:

- Host Token Ring o connessi con FDDI con una connessione Ethernet. Le MTU Token Ring (o FDDI) sui collegamenti terminali sono più grandi dell'MTU Ethernet del collegamento centrale.
- Il protocollo PPPoE (spesso utilizzato con l'ADSL) richiede un'intestazione da 8 byte. Ciò riduce l'MTU effettiva Ethernet a 1492 (1500 - 8).

Anche i protocolli di tunnel come GRE, IPv4sec e L2TP hanno bisogno di spazio per le rispettive intestazioni e sequenze finali. Ciò riduce anche la MTU effettiva dell'interfaccia in uscita.

Tunnel

Un tunnel è un'interfaccia logica usata sui router Cisco per incapsulare i pacchetti in un protocollo di trasporto.

Questa architettura è stata progettata per fornire i servizi che consentono di implementare uno schema di incapsulamento point-to-point. Le interfacce tunnel hanno tre componenti principali:

- Protocollo passeggeri (AppleTalk, Banyan VINES, CLNS, DECnet, IPv4 o IPX)
- Protocollo vettore: uno dei seguenti protocolli di incapsulamento:
  - GRE - Cisco multiprotocol carrier protocol. Per ulteriori informazioni, vedere la RFC 2784 e la RFC 1701.
  - IPv4 nei tunnel IPv4: per ulteriori informazioni, vedere la RFC 2003.
- Protocollo di trasporto: il protocollo usato per trasportare il protocollo incapsulato.

I pacchetti mostrati in questa sezione spiegano i concetti del tunneling IPv4 con protocollo di incapsulamento GRE e protocollo di trasporto IPv4.

Il protocollo passeggeri è sempre IPv4. In questo caso, l'IPv4 è sia protocollo di trasporto sia protocollo passeggeri.

Pacchetto normale

IPv4	TCP	Telnet

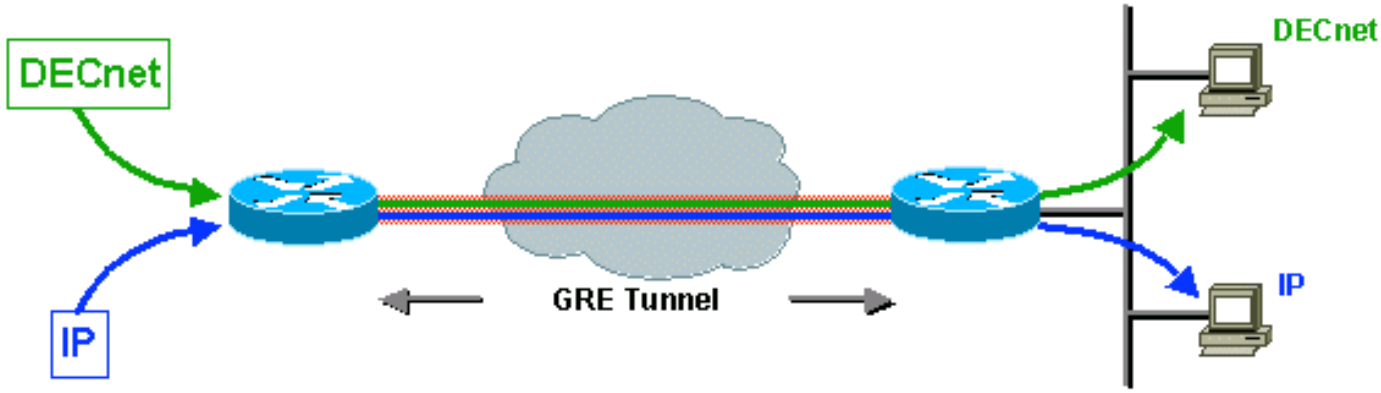
Packet tunnel

IPv4	GRE	IPv4	TCP	Telnet

- IPv4 è il protocollo di trasporto.
- GRE è il protocollo di incapsulamento.
- IPv4 è il protocollo passeggeri.

Nell'esempio seguente viene mostrato un incapsulamento in cui IPv4 e DECnet sono i protocolli passeggeri e GRE il protocollo vettore.

Questa opzione mostra la possibilità che i protocolli vettore incapsulino più protocolli passeggeri, come mostrato nell'immagine.



Si prende in considerazione il tunneling in una situazione in cui vi sono due reti non IPv4 non contigue separate da una backbone IPv4.

Se le reti non contigue eseguono DECnet, l'amministratore può scegliere di connetterle tra loro (o non connetterle) configurando DECnet nella backbone.

Inoltre, si desidera evitare che il routing DECNet usi la larghezza di banda della backbone in quanto ciò potrebbe ridurre le prestazioni della rete IPv4.

Un'alternativa valida è eseguire il tunneling del DECnet sulla backbone IPv4. La soluzione del tunnel incapsula i pacchetti DECnet all'interno del protocollo IPv4 e li invia tramite la backbone all'endpoint del tunnel dove l'incapsulamento viene rimosso e i pacchetti DECnet vengono indirizzati alla destinazione finale tramite DECnet.

L'incapsulamento del traffico all'interno di un altro protocollo offre i seguenti vantaggi:

- Gli endpoint utilizzano indirizzi privati (RFC 1918) e la backbone non supporta il routing di questi indirizzi.
- Permette di usare le VPN (Virtual Private Network) sulle WAN o su Internet.
- Unisce le reti multiprotocollo non contigue su una backbone a protocollo singolo.
- Cripta il traffico sulla backbone o su Internet.

Di seguito, il protocollo IPv4 viene usato come protocollo passeggeri e come protocollo di

trasporto.

Considerazioni sulle interfacce tunnel

Ecco alcune considerazioni relative al tunneling.

- La commutazione rapida dei tunnel GRE è stata introdotta in Cisco IOS ® versione 11.1, la commutazione CEF è stata introdotta nella versione 12.0.
- La commutazione CEF dei tunnel GRE multipoint è stata introdotta nella versione 12.2(8)T.
- Nelle versioni precedenti di Cisco IOS ®, dove era supportata solo la commutazione di contesto, le operazioni di incapsulamento e decapsulamento erano lente.
- Quando si usa il tunneling dei pacchetti occorre essere consapevoli dei problemi di sicurezza e topologia che comporta. I tunnel possono ignorare gli Access Control Lists (ACL) e i firewall.
- Se si crea un tunnel attraverso il firewall, il protocollo passeggeri da tunneling verrà ignorato. Pertanto, si consiglia di includere la funzionalità firewall sugli endpoint del tunnel per applicare eventuali policy sui protocolli passeggeri.
- Il tunneling crea problemi con i protocolli di trasporto dotati di timer limitati (ad esempio, DECnet) a causa di una maggiore latenza.
- Il tunneling in ambienti con collegamenti di diversa velocità, come le reti ad anello FDDI veloci e le linee telefoniche lente a 9600 bps, introduce problemi di riordino dei pacchetti. Alcuni protocolli passeggeri funzionano male nelle reti miste.
- I tunnel point-to-point utilizzano la larghezza di banda su un collegamento fisico. Su più tunnel point-to-point, ogni interfaccia del tunnel ha una sua larghezza di banda così come l'interfaccia fisica su cui è in esecuzione il tunnel. Ad esempio, impostare la larghezza di banda del tunnel a 100 Kb con 100 tunnel in esecuzione su un collegamento a 10 Mb. La larghezza di banda predefinita per un tunnel è 9 Kb.
- I protocolli di routing preferiscono il tunnel di un collegamento reale, perché il tunnel sembra all'apparenza un collegamento con un solo hop, e quindi il percorso più economico, anche se comporta più hop e quindi più costoso. Per evitare questo problema, occorre configurare correttamente il protocollo di routing. Prendere in considerazione l'uso di due protocolli di routing distinti per l'interfaccia del tunnel e per l'interfaccia fisica.
- Per evitare problemi di routing ricorsivo, è possibile configurare i percorsi statici appropriati per la destinazione del tunnel. Il percorso ricorsivo si ha quando il miglior percorso per raggiungere la destinazione del tunnel è usare il tunnel stesso. In questo caso, l'interfaccia del tunnel è instabile. Questo errore si verifica quando è presente un problema di routing ricorsivo.

```
%TUN-RECURDOWN Interface Tunnel 0
```

```
temporarily disabled due to recursive routing
```

Ruoli svolti dal router durante il processo PMTUD sull'endpoint del tunnel

Quando è l'endpoint di un tunnel, il router svolge due ruoli PMTUD distinti.

- Nel primo ruolo, il router è il mittente di un pacchetto host. Per il processo PMTUD, il router deve controllare il bit DF e le dimensioni del pacchetto dati originale e adottare le misure appropriate quando necessario.
- Dopo aver incapsulato il pacchetto IPv4 originale nel pacchetto del tunnel, entra in gioco il secondo ruolo. In questa fase, il router agisce più come un host rispetto al processo PMTUD e al pacchetto IPv4 del tunnel.

Quando il router svolge il primo ruolo, ossia inoltra i pacchetti IPv4 dell'host, svolge questa attività prima di incapsulare il pacchetto IPv4 dell'host nel pacchetto del tunnel.

Quando il router è il mittente di un pacchetto host, svolge le seguenti attività:

- Controlla se il bit DF è impostato.
- Verifica quali dimensioni del pacchetto possono essere ospitate nel tunnel.
- Esegue la frammentazione (se il pacchetto è troppo grande e il bit DF non è impostato), incapsula i frammenti e li invia; oppure
- Elimina il pacchetto (se il pacchetto è troppo grande e il bit DF è impostato) e invia un messaggio ICMP al mittente.
- Incapsula il pacchetto (se non è troppo grande) e lo invia.

In genere, la scelta è tra incapsulare e frammentare (con l'invio di due frammenti incapsulati) o frammentare e incapsulare (invio di due frammenti incapsulati).

In questa sezione, illustreremo due esempi che mostrano l'interazione tra il PMTUD e i pacchetti che attraversano le reti.

Nel primo esempio viene mostrato ciò che accade a un pacchetto quando il router (all'origine del tunnel) svolge il ruolo di router di inoltro.

Per elaborare il PMTUD, il router deve verificare il bit DF e le dimensioni del pacchetto dati originale e adottare le misure appropriate.

In questo esempio viene usato l'incapsulamento GRE per il tunnel. GRE esegue la frammentazione prima dell'incapsulamento.

Negli scenari mostrati di seguito invece, l'incapsulamento precede la frammentazione.

Nell'esempio 1, il bit DF non è impostato (DF = 0) e l'MTU del protocollo IPv4 del tunnel GRE è

1476 (1500 - 24).

Esempio 1

1. Il router di inoltro (all'origine del tunnel) riceve un datagramma di 1500 byte, il cui bit DF non è impostato (DF = 0), dall'host di invio.

Questo datagramma è composto da un'intestazione IP di 20 byte e da un payload TCP di 1480 byte.

IPv4	1480 byte - TCP + dati

2. Poiché il pacchetto è troppo grande per l'MTU IPv4 dopo l'aggiunta del GRE (24 byte), il router di inoltro suddivide il datagramma in due frammenti di 1476 (intestazione IPv4 da 20 byte + payload IPv4 da 1456 byte) e 44 byte (20 byte di intestazione IPv4 + 24 byte di payload IPv4)

Dopo aver aggiunto l'incapsulamento GRE, il pacchetto non è più grande dell'MTU dell'interfaccia fisica in uscita.

IP_0	1456 byte - TCP + dati
IP_1	24 byte - dati

3. Il router di inoltro aggiunge l'incapsulamento GRE, con un'intestazione GRE da 4 byte e un'intestazione IPv4 da 20 byte, a ciascun frammento del datagramma IPv4 originale.

Questi due datagrammi IPv4 ora hanno una lunghezza di 1500 e 68 byte e sono trattati come singoli datagrammi IPv4, non come frammenti.

IPv4	GRE	IP_0	1456 byte - TCP + dati
IPv4	GRE	IP_1	24 byte - dati

4. Il router di destinazione del tunnel rimuove l'incapsulamento GRE da ciascun frammento del datagramma originale, lasciando due frammenti IPv4 lunghi 1476 e 24 byte.

Questi frammenti del datagramma IPv4 vengono inoltrati separatamente dal router diretti all'host ricevente.

IP_0	1456 byte - TCP + dati
IP_1	24 byte - dati

5. L'host ricevente ricompone i due frammenti nel datagramma originale.

IPv4	1480 byte - TCP + dati

Nell'esempio 2 viene illustrato il ruolo del router di inoltro nel contesto di una topologia di rete.

Il router svolge sempre il ruolo di router di inoltro, ma questa volta il bit DF è impostato (DF = 1).

Esempio 2

1. Il router di inoltro all'origine del tunnel riceve un datagramma di 1500 byte con DF = 1 dall'host di invio.

IPv4	1480 byte - TCP + dati

2. Poiché il bit DF è impostato e le dimensioni del datagramma (1500 byte) sono maggiori della MTU del protocollo IPv4 del tunnel GRE (1476), il router scarta il datagramma e invia un messaggio ICMP "frammentazione richiesta ma bit DF impostato" alla sorgente del datagramma.

Il messaggio ICMP avvisa il mittente che l'MTU è 1476.

IPv4	ICMP MTU 1476

3. L'host di invio riceve il messaggio ICMP e, quando invia nuovamente i dati originali, usa un datagramma IPv4 da 1476 byte.

IPv4	1456 byte - TCP + dati

4. La lunghezza del datagramma IPv4 (1476 byte) è ora uguale al valore dell'MTU IPv4 del tunnel GRE, il router può aggiungere l'incapsulamento GRE al datagramma IPv4.

IPv4	GRE	IPv4	1456 byte - TCP + dati

5. Il router ricevente (destinazione del tunnel) rimuove l'incapsulamento GRE del datagramma IPv4 e lo invia all'host ricevente.

IPv4	1456 byte - TCP + dati

Questo è quello che succede quando il router svolge il secondo ruolo, ossia il ruolo di host di invio, rispetto al PMTUD e al pacchetto IPv4 del tunnel.

Il router svolge questa attività dopo aver incapsulato il pacchetto IPv4 originale nel pacchetto del tunnel.

Nota: per impostazione predefinita, un router non esegue il PMTUD sui pacchetti del tunnel GRE che ha generato. Il comando **tunnel path-mtu-discovery** può essere usato per attivare il PMTUD sui pacchetti del tunnel GRE-IPv4.

Nell'esempio 3 viene mostrato cosa succede quando l'host invia datagrammi IPv4 delle dimensioni adatte all'MTU IPv4 sull'interfaccia del tunnel GRE.

In questo caso, il bit DF è ininfluente (1 o 0).

Sull'interfaccia del tunnel GRE il **tunnel path-mtu-discovery** comando non è configurato, quindi il router non è in grado di eseguire il PMTUD sul pacchetto GRE-IPv4.

Esempio 3

1. Il router di inoltro all'origine del tunnel riceve un datagramma di 1476 byte dall'host di invio.

IPv4	1456 byte - TCP + dati

2. Questo router incapsula il datagramma IPv4 da 1476 byte all'interno del GRE per ottenere un datagramma GRE IPv4 da 1500 byte.

Il bit DF nell'intestazione GRE IPv4 viene annullato (DF = 0). Il router inoltra quindi il pacchetto alla destinazione del tunnel.

IPv4	GRE	IPv4	1456 byte - TCP + dati

3. Si supponga che tra l'origine del tunnel e la destinazione vi sia un router con MTU del collegamento di 1400.

Il router frammenta il pacchetto del tunnel perché il bit DF non è impostato (DF = 0).

In questo esempio, viene frammentato l'IPv4 più esterno, quindi il GRE, l'IPv4 interno e le intestazioni TCP compariranno solo nel primo frammento.

IP_0	GRE	IP	1352 byte - TCP + dati
IP_1	104 byte - dati		

4. Il router di destinazione del tunnel deve ricomporre il pacchetto del tunnel GRE.

IP	GRE	IP	1456 byte - TCP + dati

5. Dopo aver ricomposto il pacchetto del tunnel GRE, il router rimuove l'intestazione GRE IPv4 e invia il datagramma IPv4 originale.

IPv4	1456 byte - TCP + dati

Nell'esempio 4 viene mostrato cosa succede quando il router svolge il ruolo di host di invio rispetto al PMTUD e al pacchetto IPv4 del tunnel.

In questo caso, il bit DF è impostato (DF = 1) nell'intestazione dell'IPv4 originale e il **tunnel path-mtu-discovery** comando è stato configurato in modo che il bit DF dell'intestazione IPv4 interna venga copiato sull'intestazione (GRE + IPv4) esterna.

Esempio 4

1. Il router di inoltro all'origine del tunnel riceve un datagramma di 1476 byte con DF = 1 dall'host di invio.

IPv4	1456 byte - TCP + dati

2. Questo router incapsula il datagramma IPv4 da 1476 byte all'interno del GRE per ottenere un datagramma GRE IPv4 da 1500 byte.

Nell'intestazione GRE IPv4, il bit DF è impostato (DF = 1), per riflettere la situazione del datagramma IPv4 originale.

Il router inoltra quindi il pacchetto alla destinazione del tunnel.

IPv4	GRE	IPv4	1456 byte - TCP

3. Si supponga inoltre che tra l'origine del tunnel e la destinazione vi sia un router con MTU del collegamento di 1400.

Questo router non frammenta il pacchetto del tunnel perché il bit DF è impostato (DF = 1).

Il router deve eliminare il pacchetto e inviare un messaggio di errore ICMP al router di origine del tunnel, perché è l'indirizzo IPv4 di origine sul pacchetto.

IPv4	ICMP MTU 1400

4. Il router di inoltro sull'origine del tunnel riceve il messaggio di errore "ICMP" e diminuisce l'MTU IPv4 del tunnel GRE a 1376 (1400 - 24).

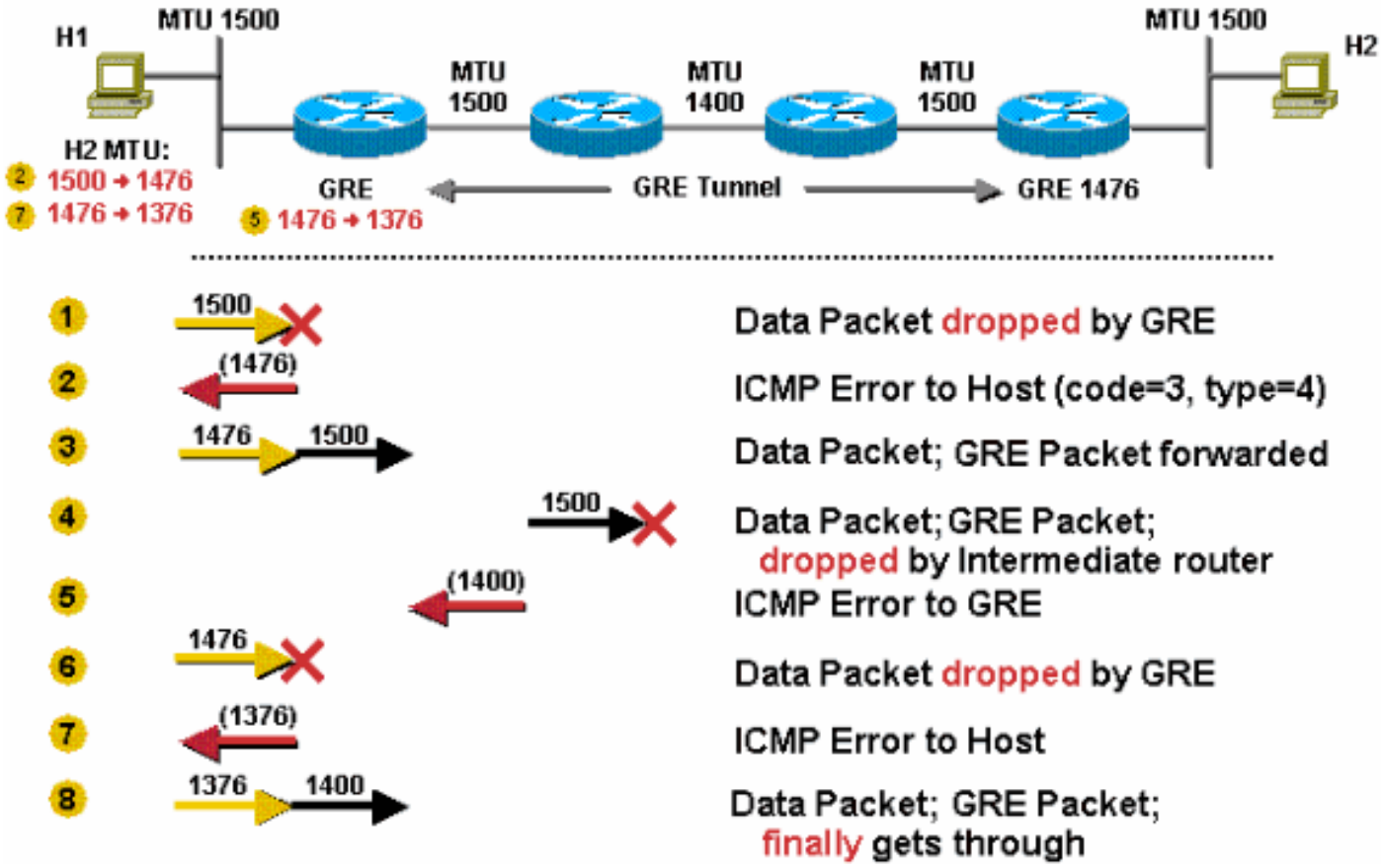
Al successivo invio dei dati in un pacchetto IPv4 da 1476 byte, il pacchetto risulta ancora troppo grande e il router invia un messaggio di errore "ICMP" al mittente con un valore MTU di 1376.

Quando l'host di invio ritrasmette i dati, li include in un pacchetto IPv4 da 1376 byte e ora questo pacchetto può passare attraverso il tunnel GRE e raggiungere l'host ricevente.

Esempio 5

Nell'esempio viene illustrata la frammentazione del GRE. Esegue la frammentazione prima dell'incapsulamento per GRE, quindi esegue il PMTUD per il pacchetto dati e il bit DF non viene copiato quando il pacchetto IPv4 viene incapsulato dal GRE.

Il bit DF non è impostato. Per impostazione predefinita, l'MTU IPv4 dell'interfaccia del tunnel GRE è più piccola dell'MTU IPv4 dell'interfaccia fisica di 24 byte, quindi è pari a 1476 come mostrato nell'immagine.



1. Il mittente invia un pacchetto da 1500 byte (20 byte di intestazione IPv4 + 1480 byte di payload TCP).
2. Poiché l'MTU del tunnel GRE è 1476, il pacchetto da 1500 byte viene suddiviso in due frammenti IPv4 di 1476 e 44 byte, ciascuno in previsione dell'intestazione GRE supplementare di 24 byte.
3. I 24 byte dell'intestazione GRE vengono aggiunti a ciascun frammento IPv4. Ora i frammenti sono rispettivamente da 1500 (1476 + 24) e da 68 (44 + 24) byte.
4. I pacchetti GRE + IPv4 che contengono i due frammenti IPv4 vengono inoltrati al router peer del tunnel GRE.
5. Il router peer del tunnel GRE rimuove le intestazioni GRE dai due pacchetti.
6. Il router inoltra i due pacchetti all'host di destinazione.
7. L'host di destinazione ricompone i frammenti IPv4 nel datagramma IPv4 originale.

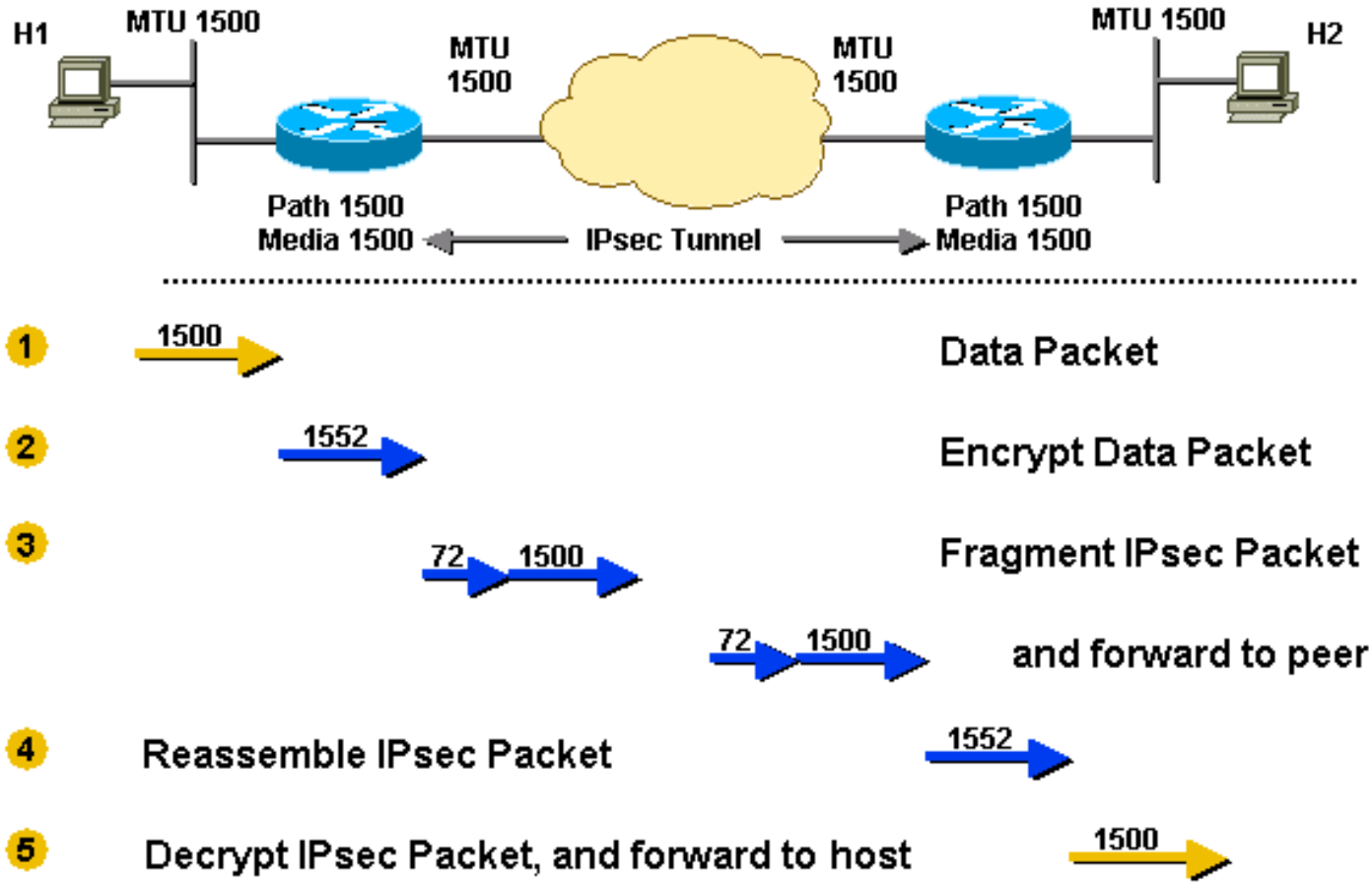
Esempio 6

Questo esempio è simile all'esempio 5, ma questa volta il bit DF è impostato. Il router è configurato in modo da eseguire il PMTUD sui pacchetti del tunnel GRE + IPv4 con il **tunnel path-mtu-discovery**comando 1. Il bit DF dell'intestazione IPv4 originale viene copiato sull'intestazione GRE IPv4.

Se il router riceve un errore ICMP per il pacchetto GRE + IPv4, riduce l'MTU IPv4 sull'interfaccia del tunnel GRE.

Per impostazione predefinita, l'MTU IPv4 del tunnel GRE è inferiore all'MTU dell'interfaccia fisica di 24 byte, quindi l'MTU del GRE IPv4 è pari a 1476. Il percorso del tunnel GRE contiene un

collegamento con MTU pari a 1400, come mostrato nell'immagine.



1. Il router riceve un pacchetto da 1500 byte (intestazione IPv4 da 20 byte + payload TCP da 1480) e scarta il pacchetto. Il router scarta il pacchetto perché è più grande dell'MTU IPv4 (1476) sull'interfaccia del tunnel GRE.
2. Il router invia un errore ICMP al mittente per comunicargli che l'MTU dell'hop successivo è 1476. L'host registra queste informazioni, generalmente come percorso host alla destinazione nella relativa tabella di routing.
3. Quando deve inviare nuovamente i dati, l'host di invio usa pacchetti da 1476 byte. Il router GRE aggiunge 24 byte di incapsulamento GRE e invia un pacchetto da 1500 byte.
4. Il pacchetto da 1500 byte non può attraversare il collegamento da 1400 byte, quindi viene scartato dal router intermedio.
5. Il router intermedio invia un messaggio ICMP (tipo = 3, codice = 4) al router GRE per comunicare che la MTU dell'hop successivo è pari a 1400. Il router GRE riduce questo valore a 1376 (1400 - 24), quindi imposta un valore MTU IPv4 interno sull'interfaccia GRE. La modifica può essere verificata solo con il comando **debug tunnel** command; non può essere verificata nell'output del **show ip interface tunnel**<#> comando command.
6. Al successivo invio di un pacchetto da 1476 byte, il router GRE lo scarta, in quanto supera le dimensioni attuali dell'MTU IPv4 (1376) sull'interfaccia del tunnel GRE.
7. Il router GRE invia un altro messaggio ICMP (tipo = 3, codice = 4) al mittente con una MTU dell'hop successivo di 1376 byte. L'host aggiorna le sue informazioni con il nuovo valore.
8. Al successivo invio, il pacchetto è ora di 1376 byte, a cui vanno aggiunti 24 byte di incapsulamento del GRE. Il pacchetto viene quindi inoltrato. Questa volta il pacchetto passa attraverso il router peer del tunnel GRE, dove il pacchetto viene decapsulato e inviato

all'host di destinazione.

tunnel path-mtu-discovery Nota: se il comando non è stato configurato sul router di inoltro di questo scenario e il bit DF è stato impostato nei pacchetti inoltrati tramite il tunnel GRE, l'host 1 riesce comunque a inviare i pacchetti TCP/IPv4 all'host 2, ma questi saranno frammentati sul collegamento centrale con MTU di 1400. Anche il peer del tunnel GRE deve ricomporre i frammenti prima di poter decapsulare il pacchetto e inoltrarlo.

Modalità tunnel IPsec puro

Il protocollo IPv4 Security (IPv4sec) è un metodo basato su standard che fornisce privacy, integrità e autenticità alle informazioni trasmesse sulle reti IPv4.

IPv4sec fornisce la crittografia IPv4 a livello di rete. Con il protocollo IPv4sec, il pacchetto IPv4 diventa più lungo perché viene aggiunta almeno un'intestazione IPv4 (modalità tunnel).

Le intestazioni aggiunte variano in lunghezza a seconda della modalità di configurazione IPv4sec, ma non superano i 58 byte circa (autenticazione Encapsulating Security Payload (ESP) ed ESP (ESPauth)) per pacchetto.

IPv4sec può funzionare in due modalità, la modalità tunnel e la modalità trasporto.

1. La modalità tunnel è la modalità predefinita. Nella modalità tunnel, l'intero pacchetto IPv4 originale viene protetto (criptato, autenticato o entrambi) e incapsulato dalle intestazioni e dalle sequenze terminali di IPv4sec. Quindi, una nuova intestazione IPv4 viene anteposta al pacchetto, per distinguere gli endpoint IPv4sec (peer) di origine e destinazione. La modalità tunnel può essere utilizzata con qualsiasi traffico IPv4 unicast e deve essere utilizzata se IPv4sec protegge il traffico proveniente dagli host dietro i peer IPv4sec. Ad esempio, la modalità tunnel viene usata sulle reti VPN (Virtual Private Network), in cui gli host di una rete protetta inviano pacchetti agli host di un'altra rete protetta tramite una coppia di peer IPv4sec. Sulle VPN, il "tunnel" IPv4sec protegge il traffico IPv4 tra gli host criptando i dati scambiati tra i router peer IPv4sec.
2. Nella modalità trasporto (configurata con il sottocomando **mode transport**, nella definizione di trasformazione), viene protetto solo il payload del pacchetto IPv4 originale (criptato, autenticato o entrambi). Il payload è incapsulato nelle intestazioni e nelle sequenze terminali dell'IPv4sec. Le intestazioni IPv4 originali rimangono invariate, ad eccezione del campo del protocollo IPv4 che viene cambiato in ESP (50), mentre il valore del protocollo originale viene salvato nella sequenza terminale dell'IPv4sec in modo da essere recuperato quando il pacchetto viene decriptato. La modalità di trasporto viene usata solo quando il traffico IPv4 da proteggere è il traffico tra i peer IPv4sec stessi, gli indirizzi IPv4 di origine e destinazione sul pacchetto sono gli stessi degli indirizzi peer IPv4sec. Normalmente la modalità di trasporto IPv4sec viene usata solo quando si usa un altro protocollo di tunneling, ad esempio il protocollo GRE, per incapsulare il pacchetto dati IPv4. Il protocollo IPv4sec viene usato successivamente per proteggere i pacchetti del tunnel GRE.

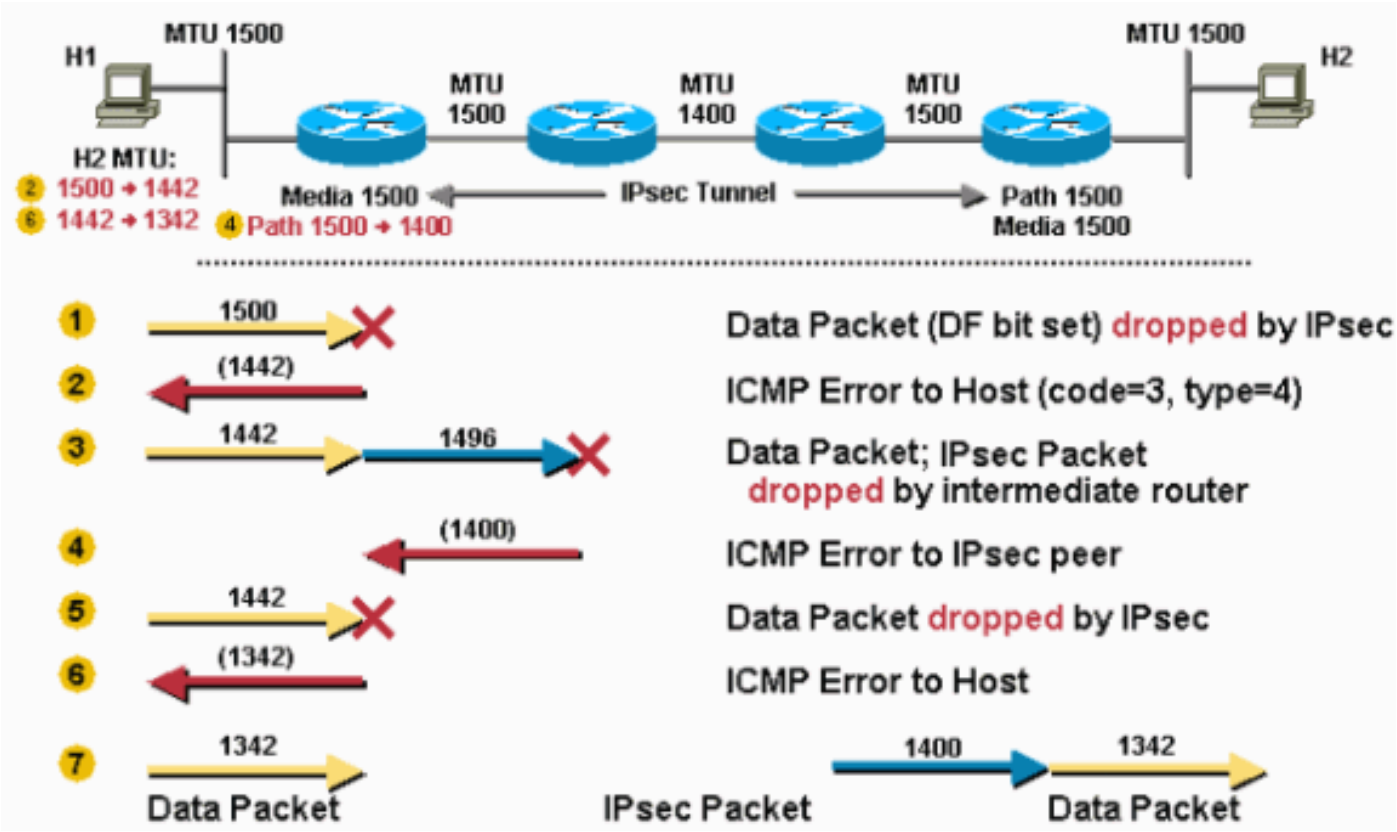
IPv4sec esegue sempre il PMTUD sui pacchetti dati e sui propri pacchetti. Per modificare la

funzionalità PMTUD sui pacchetti IPv4sec IPv4, sono disponibili alcuni comandi di configurazione, per cancellare, impostare o copiare il bit DF dell'intestazione IPv4 del pacchetto dati sull'intestazione IPv4sec IPv4. Questa funzione è denominata "funzionalità di sostituzione del bit DF".

Nota: quando si criptano i dati con IPv4sec, evitare la frammentazione dopo l'incapsulamento. La crittografia hardware offre velocità di trasmissione di circa 50 Mbs, ma se il pacchetto IPv4sec viene frammentato, la velocità può diminuire del 50-90%. Questa perdita di velocità è causata dalla commutazione di contesto, necessaria per ricomporre i pacchetti IPv4sec frammentati, che vengono quindi consegnati al motore di crittografia hardware per essere decriptati. La crittografia hardware può finire quindi per equiparare le prestazioni della crittografia software (2-10 Mbs).

Esempio 7

In questo scenario viene illustrata la frammentazione IPv4sec. L'MTU rimane 1500 per tutto il percorso e In questo scenario, il bit DF non è impostato.



1. Il router riceve un pacchetto da 1500 byte (intestazione IPv4 da 20 byte + payload TCP da 1480 byte) destinato all'host 2.
2. Il pacchetto da 1500 byte viene criptato da IPv4sec con l'aggiunta di ulteriori 52 byte (intestazione IPv4sec, sequenza terminale e intestazione IPv4 aggiuntiva). Ora IPv4sec deve inviare un pacchetto da 1552 byte. Poiché l'MTU in uscita è 1500, questo pacchetto deve essere frammentato.
3. Il pacchetto IPv4sec viene suddiviso in due frammenti. Durante la frammentazione, viene

aggiunta un'intestazione IPv4 da 20 byte al secondo frammento. I frammenti risultanti sono dunque uno da 1500 byte e uno IPv4 da 72 byte.
4. Il router peer del tunnel IPv4sec riceve i frammenti, rimuove l'intestazione IPv4 aggiuntiva e riunisce i frammenti IPv4 ricomponendo il pacchetto IPv4sec originale. Quindi, IPv4sec decripta il pacchetto.
5. Infine, il router inoltra il pacchetto dati originale da 1500 byte all'host 2.

Esempio 8

Questo esempio è simile all'esempio 6, con la differenza che in questo caso il bit DF è impostato nel pacchetto dati originale e tra i peer del tunnel IPv4sec è presente un collegamento la cui MTU è inferiore rispetto agli altri collegamenti.

In questo esempio viene mostrato come il router peer IPv4sec svolge entrambi i ruoli PMTUD, come descritto nella sezione Ruoli svolti dal router durante il processo PMTUD sull'endpoint del tunnel.

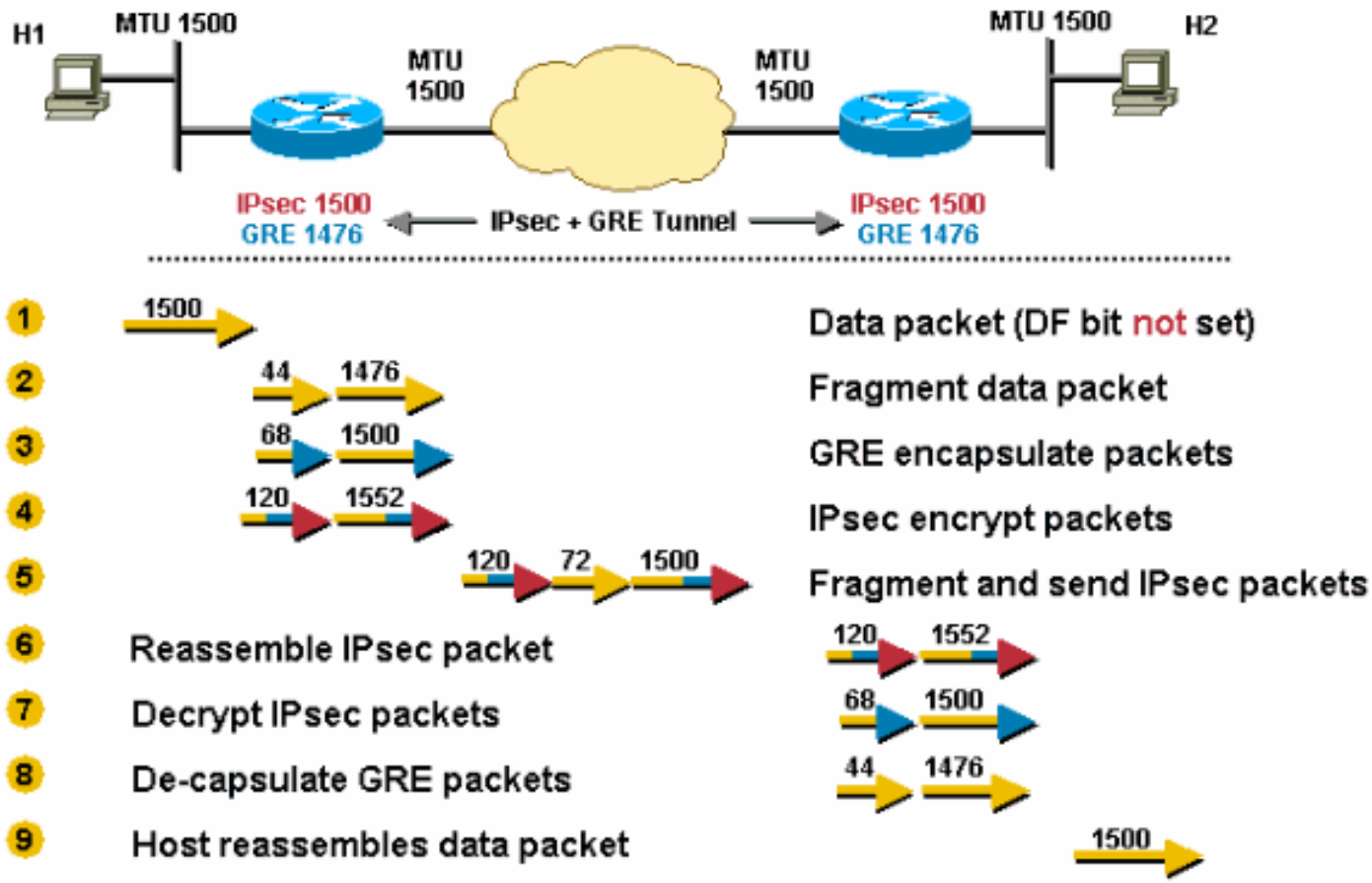
A causa della necessità di frammentare il pacchetto, la PMTU dell'IPv4sec deve essere ridotta a un valore inferiore.

Quando il protocollo IPv4sec cripta un pacchetto, il bit DF dell'intestazione IPv4 interna viene copiato sull'intestazione IPv4 esterna.

I valori MTU e PMTU medi vengono memorizzati nella Security Association (SA) dell'IPv4sec.

La MTU media si basa sulla MTU dell'interfaccia del router in uscita, mentre la PMTU si basa sulla MTU minima rilevata sul percorso tra i peer IPv4sec.

IPv4sec incapsula/cripta il pacchetto prima di tentare di frammentarlo, come mostrato nell'immagine.



1. Il router riceve un pacchetto da 1500 byte e lo scarta in quanto il sovraccarico IPv4sec, quando aggiunto, lo rende più grande della PMTU (1500).
2. Il router invia un messaggio ICMP all'host 1 per comunicare che l'MTU dell'hop successivo è 1442 (1500 - 58 = 1442). I 58 byte sono il massimo sovraccarico IPv4sec quando si usano IPv4sec ESP ed ESPauth. Il carico IPv4sec effettivo è probabilmente inferiore di 7 byte. L'host 1 registra queste informazioni, in genere come percorso host della destinazione (host 2), nella relativa tabella di routing.
3. L'host 1 riduce la PMTU dell'host 2 a 1442, quindi l'host 1, al successivo invio dei dati all'host 2, invia pacchetti più piccoli (1442 byte). Il router riceve il pacchetto da 1442 byte e IPv4sec aggiunge 52 byte di sovraccarico di crittografia, per un pacchetto IPv4sec risultante di 1496 byte. Poiché il bit DF è impostato nell'intestazione del pacchetto, il pacchetto viene scartato dal router intermedio il cui collegamento ha una MTU di 1400 byte.
4. Il router intermedio che ha scartato il pacchetto invia un messaggio ICMP al mittente del pacchetto IPv4sec (il primo router) per comunicare che l'MTU dell'hop successivo è 1400 byte. Questo valore viene registrato nella PMTU della SA IPv4sec.
5. Al successivo invio, l'host 1 trasmette un pacchetto da 1442 byte, in quanto non ha ancora ricevuto conferma, e il IPv4sec rifiuta il pacchetto. Il router scarta il pacchetto perché il sovraccarico IPv4sec, quando aggiunto al pacchetto, lo rende più grande della PMTU (1400).
6. Il router invia un messaggio ICMP all'host 1 per comunicare che l'MTU dell'hop successivo è ora 1342. (1400 - 58 = 1342). L'host 1 registra nuovamente queste informazioni.
7. Quando l'host 1 ritrasmette nuovamente i dati, utilizza il pacchetto con le dimensioni inferiori (1342). Questo pacchetto non deve essere frammentato e può passare attraverso il tunnel IPv4sec per raggiungere l'host 2.

Uso congiunto di GRE e IPv4sec

Quando si usa il protocollo IPv4sec per criptare i tunnel GRE, le operazioni di frammentazione e PMTUD si fanno più complesse.

IPv4sec e GRE vengono usati insieme perché IPv4sec non supporta pacchetti IPv4 multicast, ossia non permette di eseguire un protocollo di routing dinamico sulla rete VPN IPv4sec.

I tunnel GRE supportano il multicast, pertanto è possibile utilizzare un tunnel GRE per incapsulare il pacchetto multicast del protocollo di routing dinamico in un pacchetto GRE IPv4 unicast che può quindi essere criptato da IPv4sec.

Quando si esegue questa operazione, il protocollo IPv4sec viene spesso implementato in modalità trasporto sul GRE in quanto i peer IPv4sec coincidono con gli endpoint del tunnel GRE e la modalità trasporto permette di evitare il sovraccarico di 20 byte di IPv4sec.

Un caso interessante da esaminare è quello di un pacchetto IPv4 suddiviso in due frammenti e incapsulato dal GRE.

In questo caso IPv4sec vede due pacchetti GRE + IPv4 indipendenti. In una configurazione predefinita spesso uno dei due pacchetti è così grande da dover essere frammentato dopo essere stato criptato.

Il peer IPv4sec deve ricomporre il pacchetto prima della decrittografia. Questa "doppia frammentazione" (una volta prima del GRE e una seconda volta dopo l'IPv4sec) sul router di invio aumenta la latenza e riduce la velocità di trasmissione.

Il riassemblaggio è a commutazione di contesto, quindi in questo caso sul router ricevente si verifica un hit della CPU.

Per evitare questa situazione, è possibile impostare il valore "ip mtu" sull'interfaccia del tunnel GRE su un valore sufficientemente basso in modo da tenere conto anche del sovraccarico dei protocolli GRE e IPv4sec (per impostazione predefinita, il valore "ip mtu" dell'interfaccia del tunnel GRE corrisponde al sovraccarico MTU - GRE dell'interaccia reale in uscita).

In questa tabella vengono elencati i valori MTU consigliati per ciascuna combinazione di tunnel e modalità che presuppone che l'interfaccia fisica in uscita abbia una MTU di 1500.

Combinazione di tunnel	MTU specifica richiesta	MTU consigliata
GRE + IPv4sec (modalità trasporto)	1440 byte	1400 byte
GRE + IPv4sec (modalità tunnel)	1420 byte	1400 byte

Nota: si consiglia un valore MTU di 1400 perché copre le combinazioni più comuni di modalità GRE + IPv4sec. Inoltre, non ci sono vantaggi apprezzabili nell'aggiungere un sovraccarico extra di 20 o 40 byte. Un unico valore è più facile da ricordare e impostare e copre quasi tutti gli scenari.

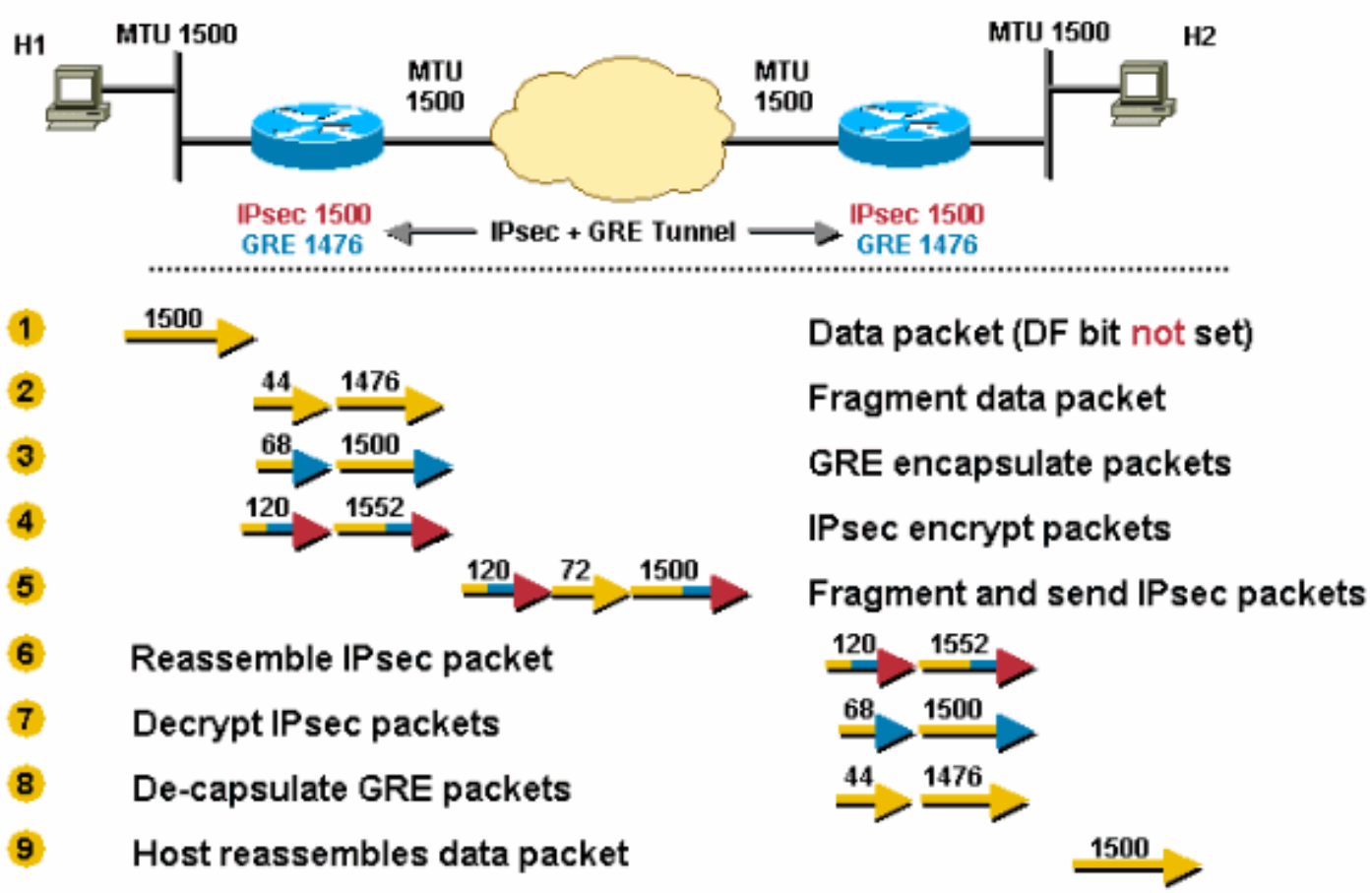
Esempio 9

Il protocollo IPv4sec viene implementato sul protocollo GRE. La MTU fisica in uscita è 1500, la PMTU dell'IPv4sec è 1500 e la MTU del GRE IPv4 è 1476 (1500 - 24 = 1476).

I pacchetti TCP/IPv4 vengono quindi frammentati due volte, una volta prima del GRE, una volta dopo l'IPv4sec.

Il pacchetto viene frammentato prima dell'incapsulamento GRE e uno di questi pacchetti GRE viene frammentato di nuovo dopo la crittografia IPv4sec.

Configurando "ip mtu 1440" (modalità trasporto IPv4sec) o "ip mtu 1420" (modalità tunnel IPv4sec) sul tunnel GRE, è possibile evitare la doppia frammentazione in questo scenario.



1. Il router riceve un datagramma di 1500 byte.
2. Prima dell'incapsulamento, GRE suddivide il pacchetto da 1500 byte in due parti, un frammento da 1476 byte (1500 - 24 = 1476) e uno da 44 byte (24 per i dati + 20 per l'intestazione IPv4).
3. GRE incapsula i frammenti IPv4, aggiungendo 24 byte a ciascun pacchetto. Il risultato sono due pacchetti GRE + IPv4sec di 1500 byte (1476 + 24 = 1500) e di 68 byte (44 + 24) ciascuno.
4. IPv4sec cripta i due pacchetti, che aggiungono 52 byte (modalità tunnel IPv4sec) di sovraccarico per l'incapsulamento a ciascuno di essi, al fine di fornire un pacchetto di 1552 byte e uno di 120 byte.
5. Il pacchetto IPv4sec di 1552 byte viene frammentato dal router perché è più grande dell'MTU

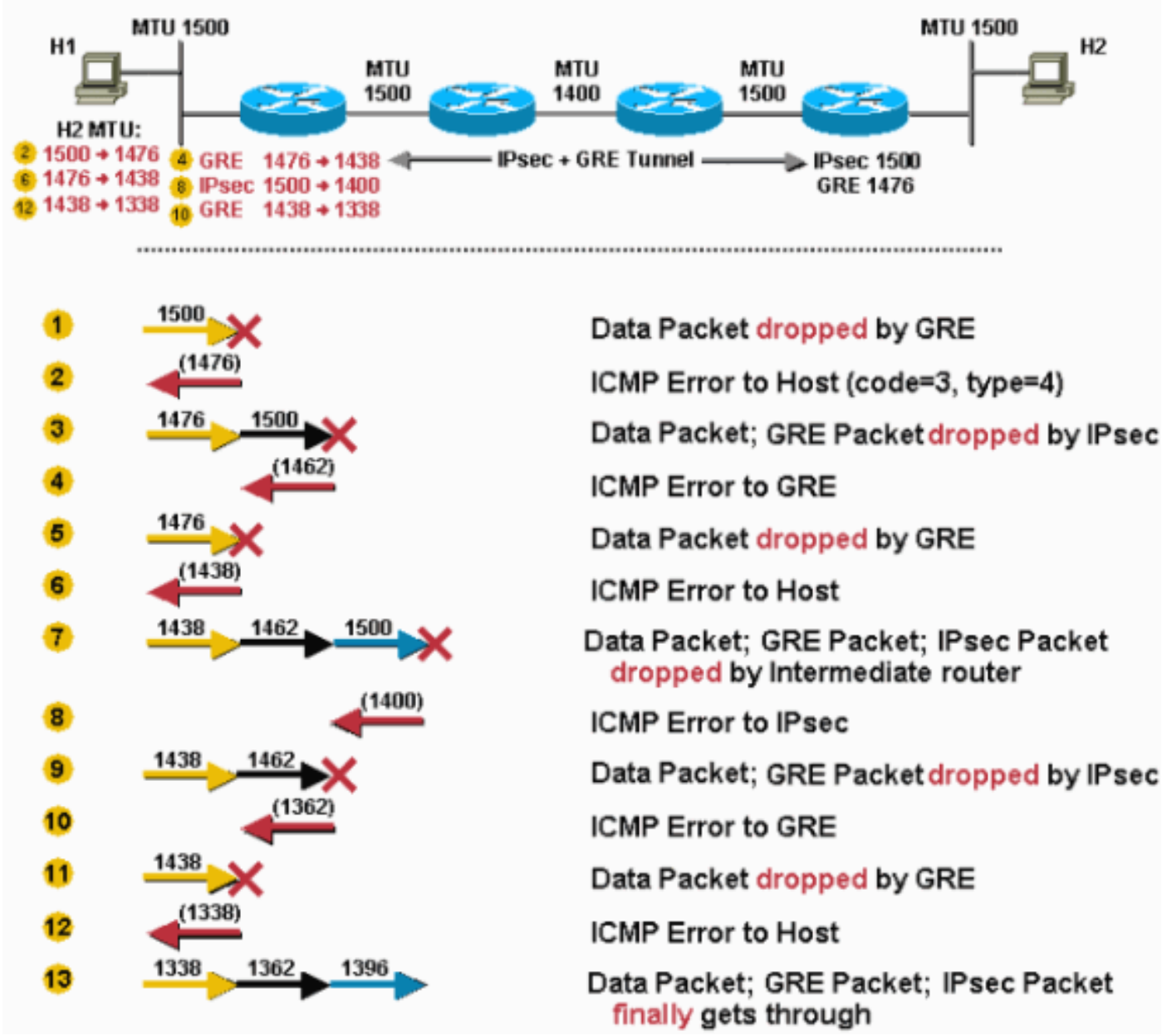
in uscita (1500). Il pacchetto di 1552 byte viene suddiviso in un pacchetto da 1500 byte e in un pacchetto da 72 byte (un payload di 52 byte più altri 20 byte per l'intestazione IPv4 del secondo frammento). I tre pacchetti da 1500 byte, 72 byte e 120 byte vengono inoltrati al peer IPv4sec + GRE.

6. Il router ricevente ricompone i due frammenti IPv4sec (da 1500 byte e 72 byte) fino a ottenere il pacchetto IPv4sec + GRE originale da 1552 byte. Il pacchetto IPv4sec + GRE da 120 byte non richiede altre operazioni.
7. IPv4sec decripta entrambi i pacchetti IPv4sec + GRE da 1552 byte e 120 byte per ottenere i pacchetti GRE da 1500 byte e 68 byte.
8. GRE estrae i pacchetti da 1500 byte e 68 byte per ottenere i frammenti IPv4 da 1476 byte e 44 byte. Questi frammenti IPv4 vengono inoltrati all'host di destinazione.
9. L'host 2 ricompone i frammenti IPv4 per ottenere il datagramma IPv4 originale da 1500 byte.

Lo scenario 10 è simile allo scenario 8, con la differenza che sul percorso del tunnel è presente un collegamento con MTU inferiore. Si tratta dello scenario peggiore per il primo pacchetto inviato dall'host 1 all'host 2. Dopo l'ultimo passaggio descritto in questo scenario, l'host 1 imposta la PMTU corretta per l'host 2 e i dati vengono scambiati correttamente sulle connessioni TCP tra i due host. I flussi TCP tra l'host 1 e gli altri host (raggiungibili con IPv4sec + tunnel GRE) devono passare attraverso le ultime tre fasi dello scenario 10.

In questo scenario, il **tunnel path-mtu-discovery** comando viene configurato sul tunnel GRE e il bit DF risulta impostato sui pacchetti TCP/IPv4 provenienti dall'host 1.

Esempio 10



- Il router riceve un pacchetto da 1500 byte. Questo pacchetto viene scartato dal GRE. Essendo il bit DF impostato a 1, il GRE non può frammentare né inoltrare il pacchetto. Inoltre, le dimensioni del pacchetto superano il valore "ip mtu" dell'interfaccia in uscita dopo l'aggiunta del sovraccarico GRE (24 byte).
- Il router invia un messaggio ICMP all'host 1 per comunicargli che la MTU dell'hop successivo è 1476 (1500 - 24 = 1476).
- L'host 1 cambia la PMTU dell'host 2 a 1476 e al successivo invio del pacchetto usa le dimensioni inferiori. GRE incapsula il pacchetto e consegna il pacchetto da 1500 byte a IPv4sec. IPv4sec scarta il pacchetto. Il GRE ha copiato il bit DF (impostato a 1) dall'intestazione IPv4 interna e, con il sovraccarico dell'IPv4sec (massimo 38 byte), il pacchetto è diventato troppo grande per essere inoltrato sull'interfaccia fisica.
- IPv4sec invia un messaggio ICMP al GRE per comunicargli che la MTU dell'hop successivo è pari a 1462 byte (considerato che un massimo di 38 byte viene aggiunto per la crittografia e il sovraccarico dell'IPv4). GRE registra il valore 1438 (1462 - 24) come "ipt mtu" sull'interfaccia del tunnel.

- Nota: il nuovo valore viene memorizzato internamente e non può essere restituito dal **show ip interface tunnel**<#>comando. Per visualizzare il nuovo valore, occorre usare il **debug tunnel** comando.

- Al successivo invio del pacchetto da 1476 byte dall'host 1, il GRE lo rifiuta.
- Il router invia un messaggio ICMP all'host 1 per comunicargli che l'MTU dell'hop successivo è 1438.
- L'host 1 riduce la PMTU per l'host 2 e trasmette nuovamente un pacchetto da 1438 byte. Questa volta, il GRE accetta il pacchetto, lo incapsula e lo consegna all'IPv4sec per farlo criptare.
- Il pacchetto IPv4sec viene inoltrato al router intermedio che, avendo un'MTU dell'interfaccia in uscita di 1400, lo rifiuta.
- Il router intermedio invia un messaggio ICMP all'IPv4sec per comunicargli che l'MTU dell'hop successivo è 1400. Questo valore viene registrato dall'IPv4sec come valore PMTU della SA IPv4sec associata.
- Quando l'host 1 ritrasmette il pacchetto da 1438 byte, il GRE lo incapsula e lo passa all'IPv4sec. L'IPv4sec rifiuta il pacchetto in quanto la sua PMTU è stata cambiata in 1400.
- L'IPv4sec invia un messaggio di errore ICMP al GRE per comunicargli che l'MTU dell'hop successivo è 1362. Il GRE registra internamente il valore di 1338.
- Quando l'host 1 ritrasmette il pacchetto originale, in quanto non ha ricevuto conferma, il GRE lo rifiuta.
- Il router invia un messaggio ICMP all'host 1 per comunicargli che l'MTU dell'hop successivo è 1338 (1362 - 24 byte). L'host 1 riduce il valore PMTU dell'host 2 a 1338.
- L'host 1 ritrasmette un pacchetto da 1338 byte e questa volta il pacchetto può finalmente raggiungere l'host 2.

Ulteriori suggerimenti

Configurare il **tunnel path-mtu-discovery** comando sull'interfaccia tunnel può favorire l'interazione tra il GRE e l'IPv4sec quando questi sono configurati sullo stesso router.

Se non si configura il **tunnel path-mtu-discovery** comando, il bit DF viene sempre annullato nell'intestazione GRE IPv4.

In questo modo, il pacchetto GRE IPv4 dovrà essere frammentato anche se sull'intestazione IPv4 dei dati incapsulati il bit DF era stato impostato in modo da non consentire la frammentazione del pacchetto.

Se il **tunnel path-mtu-discovery** comando è configurato sull'interfaccia del tunnel GRE:

1. Il GRE copia il bit DF dell'intestazione IPv4 dei dati sull'intestazione GRE IPv4.
2. Se il bit DF è impostato nell'intestazione GRE IPv4 e il pacchetto è "troppo grande" dopo la crittografia IPv4sec per la MTU IPv4 sull'interfaccia fisica in uscita, IPv4sec scarta il pacchetto e avvisa il tunnel GRE di ridurne le proprie dimensioni della MTU IPv4.
3. IPv4sec esegue il PMTUD sui propri pacchetti. Se la PMTU dell'IPv4sec cambia (diminuisce), IPv4sec non avvisa immediatamente il GRE, ma se arriva un altro pacchetto

più grande, si verifica il processo illustrato alla fase 2.
4. La MTU GRE IPv4 è ora più piccola, quindi causa il rifiuto di qualsiasi pacchetto dati IPv4 il cui bit DF risulti impostato e che ora sarà troppo grande. Inoltre, invia un messaggio ICMP all'host di invio.

Il **tunnel path-mtu-discovery** comando aiuta l'interfaccia GRE a impostare dinamicamente la MTU dell'IPv4, a differenza del **ip mtu** comando che la imposta in modo statico. Si consiglia di utilizzare entrambi i comandi.

Il **ip mtu** comando è usato per creare spazio sufficiente al sovraccarico GRE e IPv4sec per l'MTU IPv4 dell'interfaccia fisica locale in uscita.

Il **tunnel path-mtu-discovery** comando permette di ridurre ulteriormente l'MTU IPv4 del tunnel GRE, in caso tra i peer IPv4sec sia presente un collegamento con MTU IPv4 inferiore.

Di seguito sono elencate alcune delle operazioni che è possibile eseguire in caso di problemi con il PMTUD in una rete in cui sono configurati i tunnel GRE + IPv4sec.

L'elenco che segue inizia con la soluzione più desiderabile.

1. Risolvere il problema con il PMTUD che non funziona; in genere il problema è causato da un router o un firewall che blocca l'ICMP.
2. Usare il **ip tcp adjust-mss** comando sulle interfacce del tunnel in modo che il router diminuisca il valore TCP MSS nel pacchetto TCP SYN. Ciò aiuta gli host terminali (il mittente TCP e il destinatario) a usare pacchetti di dimensioni così piccole che non è necessario eseguire il PMTUD.
3. Usare il routing di policy sull'interfaccia in entrata del router e configurare una mappa dei percorsi per annullare il bit DF nell'intestazione IPv4 dei dati prima che raggiunga l'interfaccia del tunnel GRE. In questo modo, il pacchetto IPv4 dei dati verrà frammentato prima dell'incapsulamento del GRE.
4. Aumentare il valore "ip mtu" sull'interfaccia del tunnel GRE in modo che equivalga all'MTU dell'interfaccia in uscita. In questo modo, il pacchetto IPv4 dei dati verrà incapsulato dal GRE senza essere prima frammentato. Il pacchetto GRE viene quindi criptato dall'IPv4sec e frammentato per uscire dall'interfaccia fisica in uscita. In questo caso non sarà necessario configurare il **tunnel path-mtu-discovery** comando sull'interfaccia del tunnel GRE. Ciò può ridurre drasticamente la velocità di trasmissione in quanto il riassemblaggio del pacchetto IPv4 sul peer IPv4sec viene effettuato in modalità di commutazione di contesto.

Informazioni correlate

- Pagina di supporto per il routing IP
- Pagina di supporto per IPSec (IP Security Protocol)
- RFC 1191 – Rilevamento dell'MTU del percorso
- RFC 1063 – Opzioni di rilevamento dell'MTU IP
- RFC 791 – Protocollo Internet
- RFC 793 – Protocollo di controllo della trasmissione
- RFC 879 – Dimensioni massime del segmento TCP e argomenti correlati

- RFC 1701 - GRE (Generic Routing Encapsulation)
- RFC 1241 – Uno schema per il protocollo di incapsulamento Internet
- RFC 2003 – Incapsulamento IP in datagrammi IP
- Documentazione e supporto tecnico – Cisco Systems

Informazioni su questa traduzione

Cisco ha tradotto questo documento utilizzando una combinazione di tecnologie automatiche e umane per offrire ai nostri utenti in tutto il mondo contenuti di supporto nella propria lingua. Si noti che anche la migliore traduzione automatica non sarà mai accurata come quella fornita da un traduttore professionista. Cisco Systems, Inc. non si assume alcuna responsabilità per l’accuratezza di queste traduzioni e consiglia di consultare sempre il documento originale in inglese (disponibile al link fornito).