GRE および IP セキュリティでの IP フラグメンテーション、MTU、MSS、および PMTUD の問題の解決

目次

概要

このドキュメントでは、IP フラグメンテーションおよびパス最大伝送ユニット ディスカバリ（PMTUD）の動作を示し、IP トンネルの異なる組み合わせとともに使用される場合の PMTUD の動作に関連するシナリオを説明します。 インターネットで現在広く普及した IP トンネルの利用により、IP フラグメンテーションおよび PMTUD に関連する問題が顕在化しています。

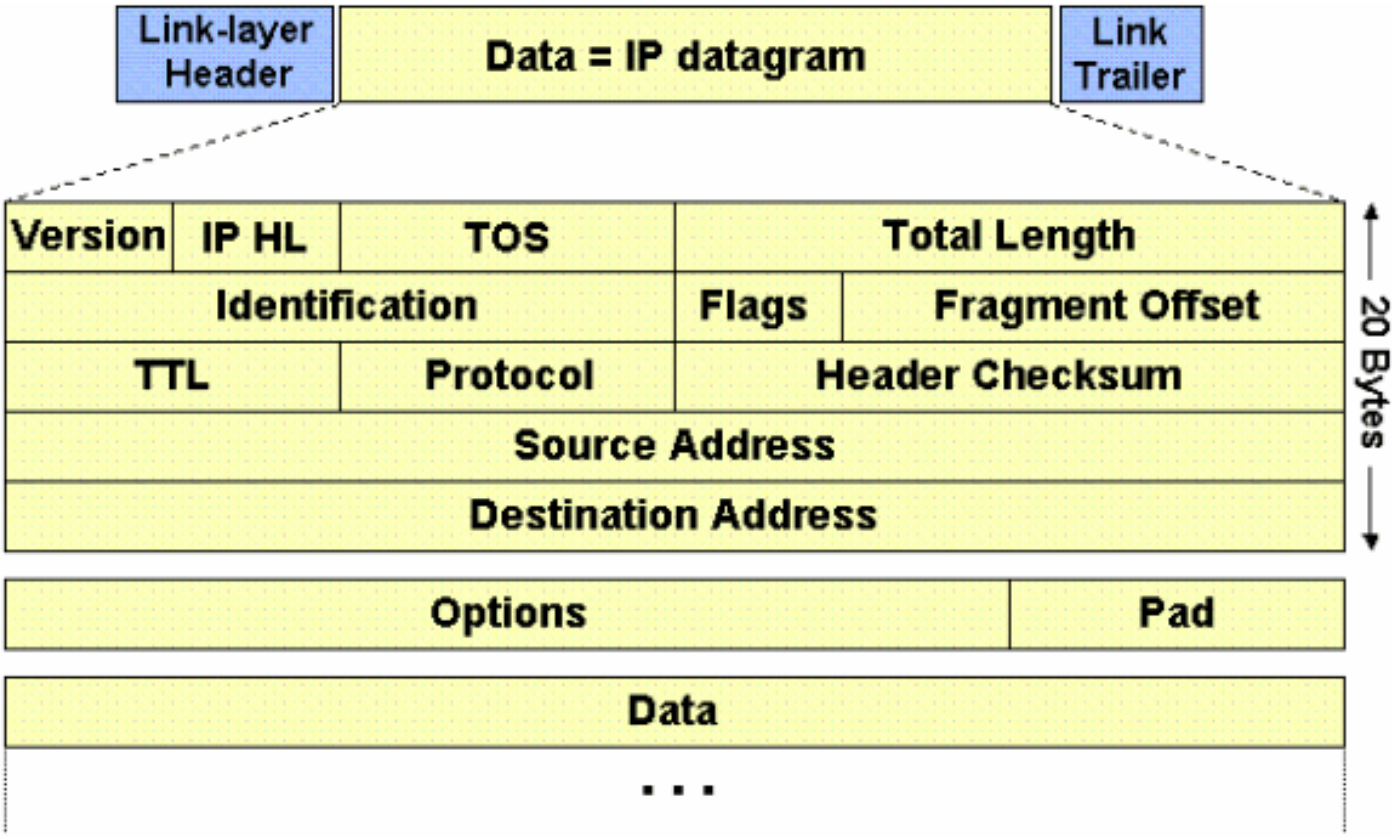
IP フラグメンテーションと再構成

IP プロトコルは、さまざまな伝送リンク上で使用する設計になっています。 IP データグラムの最大長は 65535 ですが、ほとんどの伝送リンクでは MTU と呼ばれる、より小規模な最大パケット長の制限が適用されます。 MTU の値は、伝送リンクのタイプに依存します。 IP では、ルータ

による IP データグラムのフラグメント化を必要に応じて許可することにより、異なる MTU に対応する設計になっています。 受信側ステーションには、フラグメントを元の完全なサイズの IP データグラムに再構成する役割があります。

IP フラグメンテーションでは、データグラムが、後に再構成が可能な多数の断片に分割されます。 IP ヘッダー内の「More Fragments」および「Don't Fragment」フラグとともに、IP 発信元、宛先、識別番号、合計長、およびフラグメントのオフセット フィールドが IP フラグメンテーションおよび再構成のために使用されます。 IP フラグメンテーションと再構成のしくみについての詳細は、『RFC 791』を参照してください。 。

次のイメージは IP ヘッダーのレイアウトを示しています。



識別番号は 16 ビットです。これは、データグラムのフラグメントの再構成を支援するために、IP データグラムの送信元によって割り当てられる値です。

フラグメント オフセットは 13 ビットであり、元の IP データグラムにおけるフラグメントの位置を示します。 この値は 8 バイトの倍数です。

IP ヘッダーのフラグ フィールドには、制御フラグとして 3 ビット用意されています。 「Don't Fragment」（DF）ビットによってパケットのフラグメント化を許可するかどうかが判断されるので、PMTUD ではこのビットが中心的な役割を果たすことに注意してください。

ビット 0 は予約済で、常に 0 に設定されています。ビット 1 は DF ビットです（0 =「May Fragment」、1 =「Do not Fragment」）。 ビット 2 は MF ビットです（0 =「Last Fragment」、1 =「More Fragment」）。

値	ビット 0 予約済	ビット 1 DF	ビット 2 MF
0	0	5 月	Last
1	0	Do not	More

次の図は、フラグメンテーションの例を示しています。 すべての IP フラグメントの長さを合計すると、値は元の IP データグラムの長さを 60 超過します。 全体の長さが 60 増大する理由は、最初のフラグメントの後に、3 つの追加 IP ヘッダー (各フラグメントにつき 1 つ) が作成されたからです。

最初のフラグメントはオフセット 0 であり、このフラグメントの長さは 1500 です。 これにはわずかに変更された元の IP ヘッダーの 20 バイトも含まれます。

2 番目のフラグメントのオフセットは 185 (185 x 8 = 1480) になっています。これは、このフラグメントのデータ部分が、元の IP データグラムの 1480 バイト目から始まるという意味です。 このフラグメントの長さは 1500 です。 これには、このフラグメントのために作成された追加の IP ヘッダーが含まれます。

3 番目のフラグメントのオフセットは 370 (370 x 8 = 2960) になっています。これは、このフラグメントのデータ部分が、元の IP データグラムの 2960 バイト目から始まるという意味です。 このフラグメントの長さは 1500 です。 これには、このフラグメントのために作成された追加の IP ヘッダーが含まれます。

4 番目のフラグメントのオフセットは 555 (555 x 8 = 4440) になっています。これは、このフラグメントのデータ部分が、元の IP データグラムの 4440 バイト目から始まるという意味です。 このフラグメントの長さは 700 バイトです。 これには、このフラグメントのために作成された追加の IP ヘッダーが含まれます。

元の IP データグラムのサイズを判断できるのは、最後のフラグメントが受け取られたときだけです。

最後のフラグメント (555) でのフラグメント オフセットによって、元の IP データグラムに 4440 バイトのデータ オフセット値が渡されます。 続いて、最後のフラグメントからのデータ バイトを追加すると (680 = 700 - 20)、元の IP データグラムのデータ部分である 5120 バイトが渡されたことになります。 続いて、IP ヘッダーの 20 バイトが追加され、元の IP データグラムと等しいサイズになります (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

IP フラグメンテーションに関する問題

IP フラグメンテーションで不都合が発生する、いくつかの問題があります。 IP データグラムをフラグメント化するために、CPU およびメモリ オーバーヘッドがわずかに増加します。 これは、送信側だけではなく、送信側と受信側の間のパスにおけるルータについても当てはまります。フラグメント作成では、単にフラグメント ヘッダー作成し、元のデータグラムをフラグメントにコピーするだけです。 フラグメント作成に必要なすべての情報は、ただちに利用できるので、これは非常に効率的に実行されます。

フラグメントの再構成時には、フラグメンテーションにより受信側ではそれ以上のオーバーヘッドが発生します。受信側では到着するフラグメントにメモリを割り当て、すべてのフラグメントを受け取ってから、これらを 1 つのデータグラムに結合する必要があるということが、この原因です。 ホストには、このタスクに費やす時間とメモリ リソースが備わっているため、ホスト側での再構成は問題とはなりません。

ところが、パケットをできるだけ迅速に転送することが主要な機能であるルータ上での再構成は、非常に非効率的です。 ルータは、時間にかかわらず、パケットに掛かりっきりになるようには設計されていません。 また、再構成を実行するルータでは、使用可能な最大バッファ（18 K）が処理のために選択されます。この理由は、最後のフラグメントが受け取られるまで、元の IP パケットのサイズがわからないからです。

フラグメンテーションのもう 1 つの問題は、廃棄されたフラグメントの処理方法です。 IP データグラムの 1 つのフラグメントが廃棄されると、元のすべての IP データグラムを再送する必要があり、これが再度フラグメント化されます。 Network File System（NFS; ネットワーク ファイル システム）を使用した、この例を示します。 デフォルトでは、NFS には 8192 の読み取りと書き込みのブロック サイズが用意されるので、NFS IP/UDP データグラムはおよそ 8500 バイトになります（NFS、UDP、IP ヘッダーを含む）。 イーサネット（MTU 1500）に接続された送信側ステーションでは、8500 バイトのデータグラムを 6 つ（ 5 つの 1500 バイトのフラグメントおよび 1 つの 1100 バイトのフラグメント）にフラグメント化する必要があります。 輻輳しているリンクが原因でこの 6 つのフラグメントのいずれかが廃棄された場合、元の完全なデータグラムを再転送する必要があります。つまり、さらに 6 つのフラグメントを作成する必要があります。このリンクで 6 つのパケットのうち 1 つが廃棄されるとすると、各 NFS 8500 バイトの元の IP データグラムから、少なくとも 1 つの IP フラグメントが廃棄されることになるため、このリンクを介して NFS データを転送できる可能性は低くなります。

パケット内のレイヤ 4（L4）からレイヤ 7（L7）の情報に基づいて、パケットをフィルタまたは操作するファイアウォールでは、IP フラグメントの適切な処理に失敗する場合があります。 IP フラグメントの順序が入れ替わると、ファイアウォールによって 1 番目以外のフラグメントがブロックされることがあります。この理由は、これらのフラグメントではパケット フィルタに合致する情報が伝送されないからです。 これは、受信側ホストでは元の IP データグラムの再構成が不可能であることを意味します。 不十分な情報を含む 1 番目以外のフラグメントの、フィルタへの適切な合致を許可するようにファイアウォールが設定されていると、1 番目以外のフラグメントによるファイアウォールを突破した攻撃が発生する可能性があります。 また、一部のネットワーク デバイス（コンテント スイッチ モジュール など）では、L4 から L7 の情報に基づいてパケットが操作されます。パケットが複数のフラグメントにわたる場合、このデバイスでのポリシーの実行が失敗することがあります。

IP フラグメンテーションの回避： TCP MSSが何をし、どのように動作するのか

TCP Maximum Segment Size（MSS; 最大セグメント サイズ）では、単一の TCP/IP データグラムでホストが受け取るデータの最大量が定義されます。 この TCP/IP データグラムは、IP レイヤにおいてフラグメント化される場合があります。 MSS 値は、TCP SYN セグメント内だけで

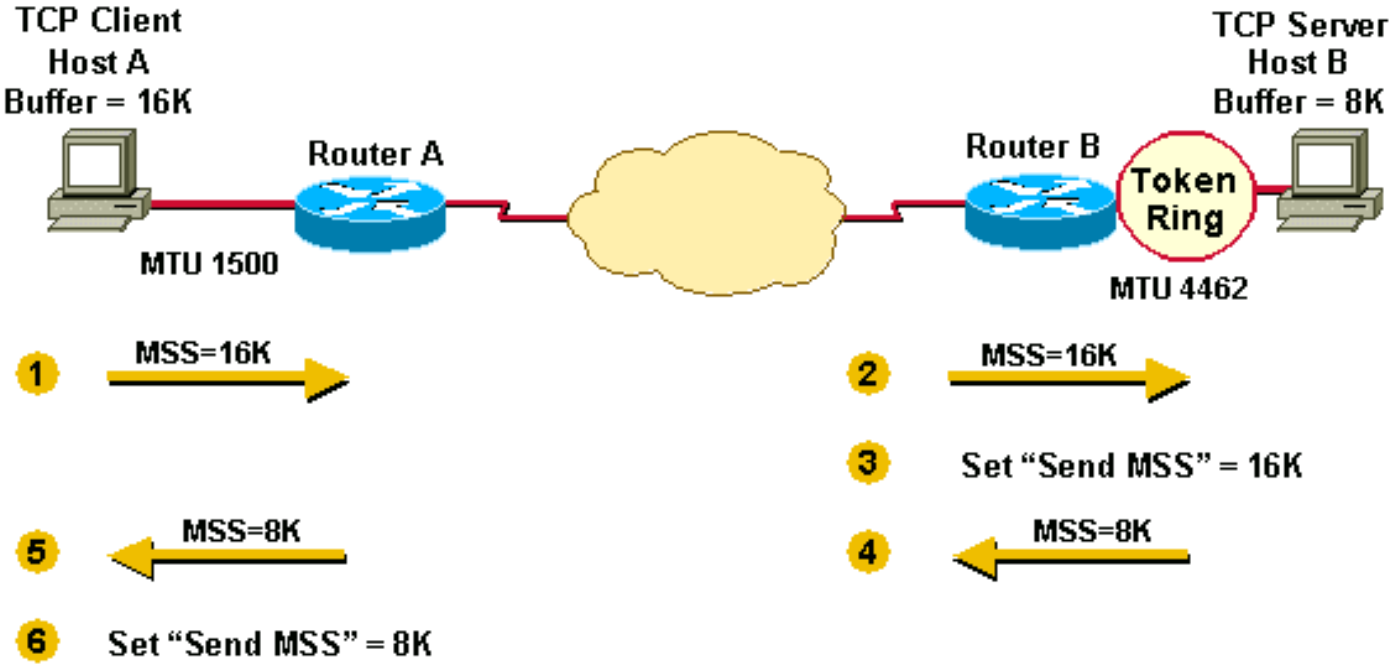
TCP ヘッダー オプションとして送信されます。 TCP 接続のそれぞれの側は、その MSS 値をもう一方の側に報告します。 一般的な認識とは異なり、ホスト間で MSS 値がネゴシエートされることはありません。 送信側ホストでは、単一の TCP セグメント内のデータ サイズを、受信側ホストから報告された MSS 以下の値に制限する必要があります。

もともと MSS とは、単一の IP データグラム内に含まれた TCP データを格納できるように、受信側ステーションに割り当てられたバッファの大きさ (65496 K 以上) を意味するものでした。つまり、MSS は TCP の受信側で受け取るデータの最大セグメント (チャンク) でした。 この TCP セグメントは、最大 64 K (最大 IP データグラム サイズ) の大きさになり、ネットワーク経由で受信側ホストに転送されるように、IP レイヤでフラグメント化することが可能です。 受信側ホストでは、IP データグラムを再構成してから、完全な TCP セグメントを TCP レイヤに渡します。

次に挙げたいくつかのシナリオでは、MSS 値を設定する方法を示します。また、MSS 値により TCP セグメントのサイズを制限することによって、IP データグラムのサイズを制限する方法も示します。

シナリオ 1 では、MSS の初期実装方法を図示します。 Host A には 16 K のバッファ、Host B には 8 K のバッファが備わっています。 これらのホスト間ではそれぞれの MSS 値が送受信され、互いのデータ送信のための送信 MSS が調整されます。 Host A および Host B では、インターフェイス MTU より大きな (ただし送信 MSS より小さい) IP データグラムをフラグメント化する必要があることに注意します。この理由は、TCP スタックにより、スタックにある 16 K または 8 K バイトのデータが IP にわたされる可能性があるからです。 Host B の場合、パケットのフラグメント化は 2 回実行されます (トークン リング LAN に到達するために 1 回、そして再度 Ethernet LAN に到達するために 1 回) 。

シナリオ 1



1. Host A は、MSS 値 16 K を Host B に送信します。
2. Host B は、Host A からの MSS 値 16 K を受信します。
3. Host B は、送信 MSS 値を 16 K に設定します。
4. Host B は、MSS 値 8K を Host A に送信します。
5. Host A は、Host B からの MSS 値 8 K を受信します。

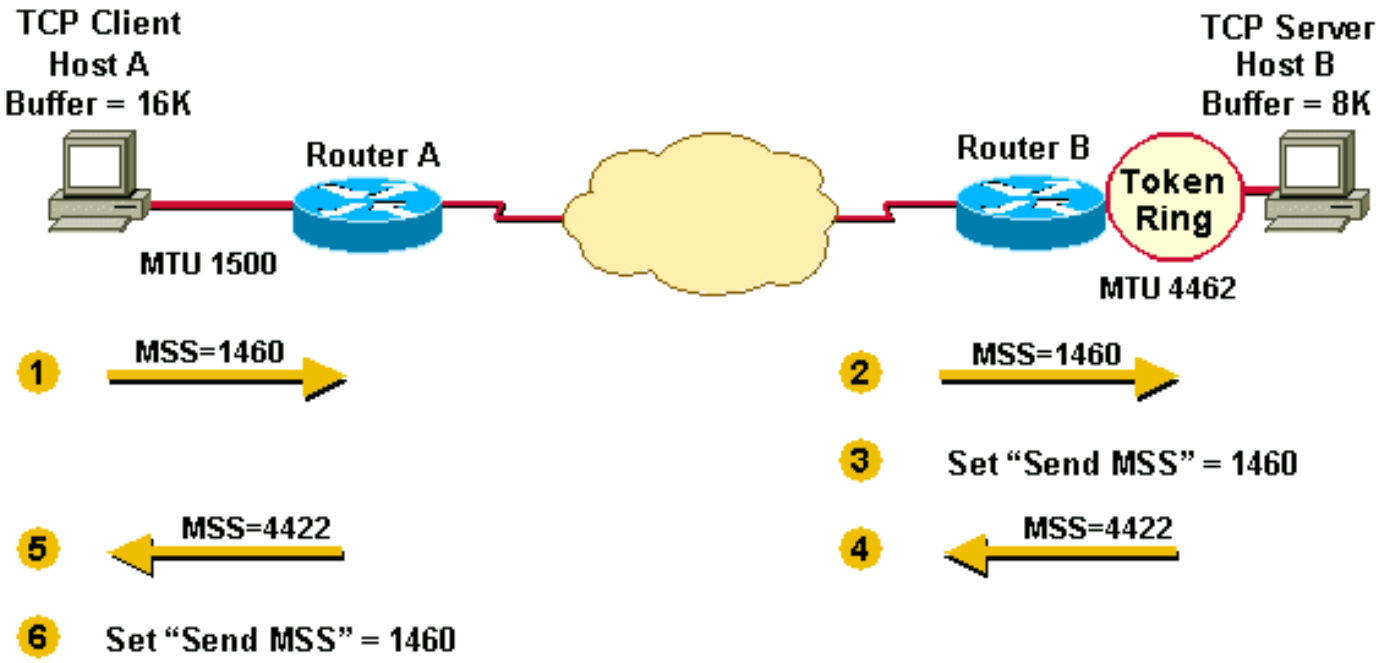
6. Host A は、送信 MSS の値を 8 K に設定します。

TCP 接続のエンドポイントでの IP フラグメンテーションの回避を支援するために、MSS 値の選択が最小バッファ サイズおよび発信インターフェイスの MTU (- 40) に変更されました。 MSS の数値は、MTU の数値より 40 バイト小さい値です。MSS が、20 バイトの IP ヘッダーと 20 バイトの TCP ヘッダーを含まない TCP データのサイズと同じであることが、この理由です。 MSS はデフォルトのヘッダー サイズに基づいています。 送信側スタックでは、使用されている TCP または IP オプションに応じて、IP ヘッダーおよび TCP ヘッダーのための適切な値を差し引く必要があります。

MSS の動作方法は次のとおりです。まず、各ホストが送信インターフェイス MTU をそれぞれのバッファと比較し、送信する MSS として最小の値を選択します。 次に、ホストは受信した MSS のサイズをそれぞれのインターフェイス MTU と再度比較し、2 つの値のより小さな値を再度選択します。

シナリオ 2 では、ローカルおよびリモート接続でのフラグメンテーションを回避するために送信側で実行されるこの追加手順を図示します。 各ホストによって (ホストが他方に MSS 値を送信する前に) どのように発信インターフェイスの MTU が考慮され、これがどのようにフラグメンテーションの回避に役立つのかに注意してください。

シナリオ2



1. Host A では、MSS バッファ (16 K) と MTU (1500 - 40 = 1460) が比較され、Host B に送信する MSS (1460) としてより低い値が使用されます。
2. Host B は、Host A の送信 MSS (1460) を受信し、これを送信インターフェイスの MTU - 40 (4422) の値と比較します。
3. Host B は、Host A に送信する IP データグラムの MSS として、より低い値 (1460) を設定します。
4. Host B は、MSS バッファ (8 K) と MTU (4462-40 = 4422) を比較し、Host A に送信する MSS として 4422 を使用します。
5. Host A は Host B の送信 MSS (4422) を受信し、これを送信インターフェイス MTU - 40 (1460) の値と比較します。
6. Host A は、Host B に送信する IP データグラムの MSS として、より低い値 (1460) を設定

します。
1460 が両方のホストによって、それぞれの送信 MSS として選択された値になります。 多くの場合、送信 MSS の値は TCP 接続の両側で同じ値になります。

シナリオ 2 では、TCP 接続のエンドポイントにおいてフラグメンテーションは発生しません。この理由は、両方の送信インターフェイス MTU がホストにより考慮されているからです。 ただし、いずれかのホストの送信インターフェイスの MTU より低い MTU を含むリンクがあると、ルータ A とルータ B の間のネットワーク内でパケットがフラグメント化される可能性があります。

PMTUD の概要

前述のように、TCP MSS は TCP 接続の 2 つのエンドポイントにおいてフラグメンテーションを取り扱いますが、これら 2 つのエンドポイントの間に、より小さな MTU のリンクが存在する場合は、この対象外となります。 エンドポイント間のパス内でのフラグメンテーションを回避するために、PMTUD が開発されました。 これは、パケットの発信元から宛先までのパス上で、最も低い MTU を動的に判断するために使用されます。

> **注:** PMTUD は TCP および UDP でのみサポートされます。 その他のプロトコルでは、これをサポートしていません。 PMTUD がホストでイネーブルになっており、ほぼ常にその状態であると、ホストからのすべての TCP/IP または UDP パケットでは DF ビットが設定されます。

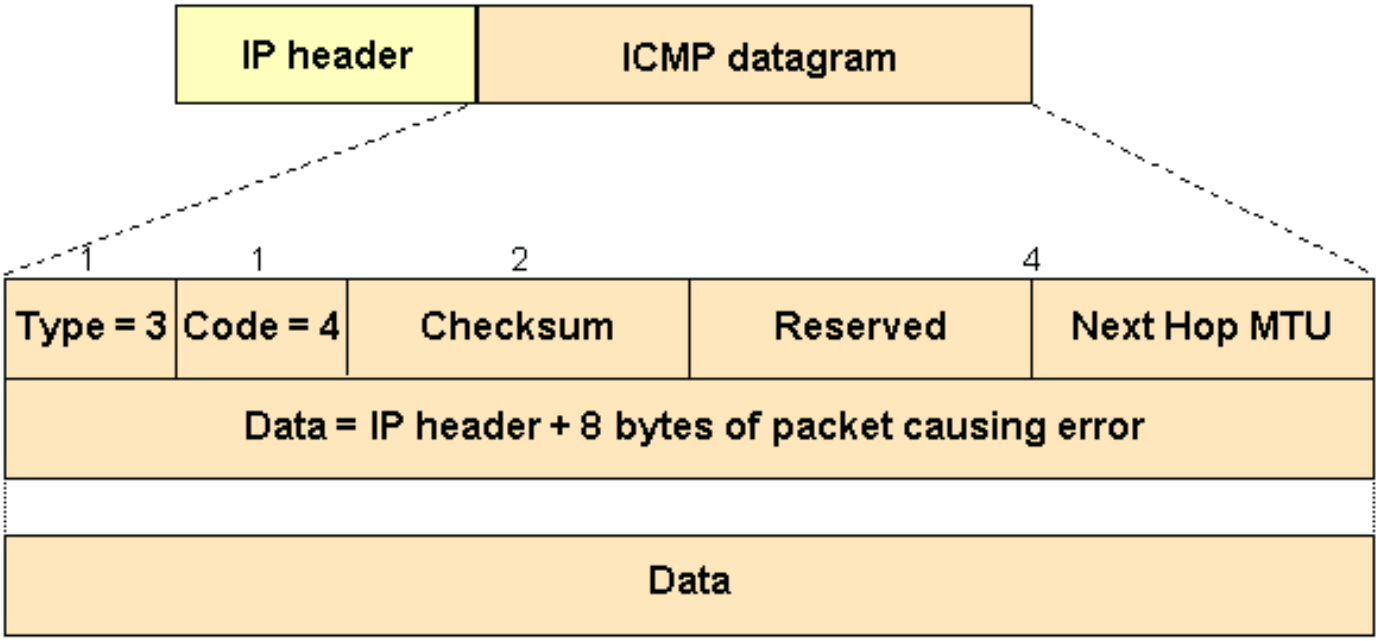
ホストが DF ビットを設定して完全な MSS データパケットを送信する際に、パケットに対してフラグメンテーションが必要であるとの情報を受け取ると、PMTUD は接続のための送信 MSS の値を低下させます。 ホストは通常、この MTU 値を含むルーティング テーブル内に「host」（/32）エントリを作成するため、宛先の MTU 値を「覚えて」います。

ルータが、パケット サイズより低い MTU を持つリンクへ、DF ビットが設定された IP データグラムの転送を試みる場合には、このルータはパケットを廃棄し、「Fragmentation Needed and DF set」（タイプ 3、コード 4）を示すコードとともに、Internet Control Message Protocol（ICMP）の「Destination Unreachable」メッセージをこの IP データグラムの発信元に返します。 発信元のステーションでは、この ICMP メッセージを受信すると、送信 MSS を低下させ、TCP がセグメントを再伝送する場合には、この小さなセグメント サイズが使用されます。

debug ip icmp コマンドを有効にした後にルータ上に表示される可能性がある、ICMP の「Fragmentation Needed and DF set」メッセージの例をここに示します。

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

次の図では、「Fragmentation Needed and DF set」および「Destination Unreachable」メッセージの ICMP ヘッダーの形式を示します。



RFC 1191 によると、「Fragmentation Ne eded and DF set」を示す ICMP メッセージを返すルータでは、ICMP 仕様の RFC 792 で「unused」とラベル付けされている ICMP 追加ヘッダー フィールド下位 16 ビット内に、ネクストホップ ネットワークの MTU が含まれる必要があります。 。

RFC 1191 の初期の実装では、ネクストホップ MTU 情報は提供されていませんでした。 この情報が提供された場合も、一部のホストではこれが無視されています。 この場合、RFC 1191 には推奨される値を掲載した表も含まれます。これらの値を参照して、PMTUD 中に MTU を低下させる必要があります。 送信 MSS の適切な値でより高速に着信させるために、これはホストによって使用されます。

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 はすべてのパケット上で継続して実行されます。この理由は、送信側と受信側の間のパスは動的に変化することがあるからです。 送信側が「Can't Fragment」ICMP メッセージを受信するたびに、ルーティング情報 (PMTUD の格納場所) が更新されます。

PMTUD 中に、次の 2 つの状態が発生する可能性があります。

- パケットがフラグメント化されずに受信側まで到着する。**注:** ルータでは、DoS 攻撃から CPU を保護するために、送信する ICMP unreachable メッセージの数が 1 秒につき 2 件に制限されます。 したがって、この状況で、1 秒につき 2 件以上 (ホストが異なる可能性がある) の ICMP メッセ時 (タイプ 3、コード 4) での応答がルータで必要だと判断されるネットワーク シナリオの場合には、**no ip icmp rate-limit unreachable [df] interface** コマンドを使用して ICMP メッセージのスロットリングを無効にできます。
- 送信側では、ICMP「Can't Fragment」メッセージを、受信側へのパス上のすべてのホップから受け取る可能性があります。

PMTUD は、TCP フローの両方向において、別々に実行されます。 フローの片方の PMTUD では、エンド ステーションによる送信 MSS の低下がトリガーされ、もう片方のエンドステーションでは元の送信 MSS が保持される場合があります。この理由は、PMTUD をトリガーするだけの大きさを持つ IP データグラムが送信されていないからです。

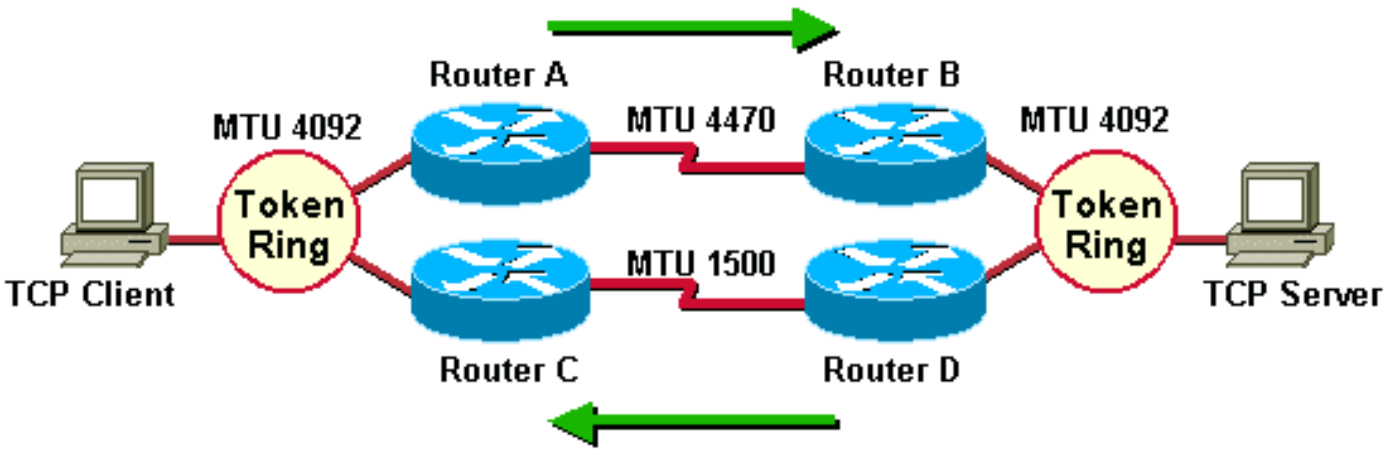
次のシナリオ 3 に図示した HTTP 接続は、このよい例です。 TCP クライアントは小さなパケットを送信し、サーバは大きなパケットを送信します。 この場合、サーバの大きなパケット（576 バイト以上）だけが PMTUD をトリガーします。 クライアントのパケットは小さく（576 バイト以下）、PMTUD をトリガーしません。この理由は、MTU が 576 のリンクを介した伝送ではフラグメンテーションが必要ではないからです。

シナリオ 3

シナリオ 4 では、パスの 1 つがその他のパスより小さな最小 MTU を持つ状況である、非対称ルーティングの例を示します。 非対称ルーティングは、2 つのエンドポイント間でのデータの送信と受信に、異なるパスが使用される場合に発生します。 このシナリオでは、TCP フローの一方向においてだけ、PMTUD によって送信 MSS の低下がトリガーされます。 TCP クライアントからサーバへのトラフィックは、ルータ A とルータ B 経由で流れますが、サーバからクライアントへのリターン トラフィックは、ルータ D とルータ C 経由で流れます。 TCP サーバによってクライアントにパケットが送信されると、PMTUD がサーバをトリガーして送信 MSS を低下させます。この理由は、ルータ C に 4092 バイトのパケットを送信する前に、ルータ D ではこれをフラグメント化する必要があるからです。

これに対して、クライアントでは、「Fragmentation Needed and DF set」を示すコードを持つ ICMP「Destination Unreachable」メッセージを受け取ることはありません。この理由は、ルータ B 経由でサーバに送信する場合、ルータ A ではパケットをフラグメント化する必要がないからです。

シナリオ 4



注: ルータ（たとえば、BGP および Telnet）によって開始された TCP 接続のための TCP

MTU パス ディスカバリを有効にするためには、ip tcp path-mtu-discovery コマンドが使用されます。

PMTUD の問題

PMTUD が失敗する 3 つの状況があります。このうちの 2 つは一般的ではありませんが、残りの 1 つは一般的です。

- ルータではパケットの破棄はできるが、ICMP メッセージが送信されない。（一般的ではない）
- ルータでは ICMP メッセージの生成と送信ができるが、ICMP メッセージが、このルータと送信側の間でのルータまたはファイアウォールによってブロックされる。（一般的）
- ルータでは ICMP メッセージの生成と送信ができるが、送信側がこのメッセージを無視する。（一般的ではない）

上記 3 項目の最初と最後の項目は一般的ではなく、通常はエラーの結果ですが、2 番目の項目は一般的な問題を示しています。 ICMP パケット フィルタを実装する担当者は、特定の ICMP メッセージ タイプだけではなく、すべての ICMP メッセージ タイプをブロックする傾向があります。 パケット フィルタでは、「unreachable」または「time-exceeded」*以外*のすべての ICMP メッセージ タイプのブロックが可能です。 PMTUD が成功したか、失敗したかは、TCP/IP パケットの送信側に到着する ICMP の unreachable メッセージで判定されます。 ICMP の time-exceeded メッセージは、その他の IP 問題にとって重要なものです。 ルータ上に実装された、このようなパケット フィルタの例を次に示します。

```
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
```

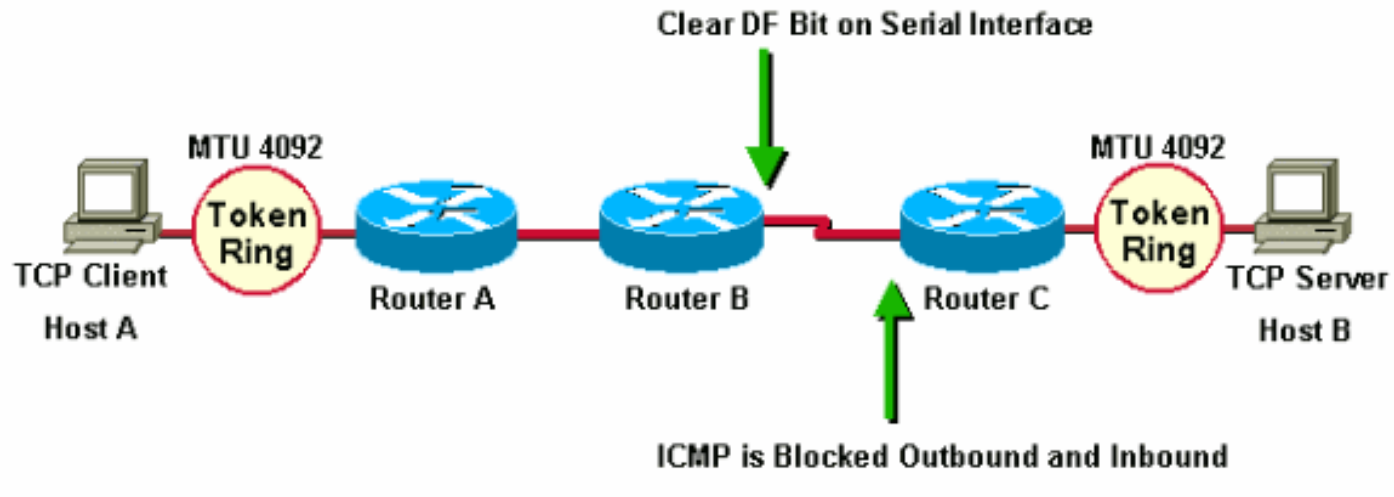
ICMP が完全にブロックされてしまうという問題の緩和に役立つ他の手法があります。

- ルータ上の DF ビットをクリアし、状況にかかわらずフラグメンテーションを許可します（ただし、これはお勧めできません。 詳細については、『IP フラグメンテーションに関する問題』を参照してください）。
- インターフェイス コマンド **ip tcp adjust-mss <500-1460>** で、TCP MSS のオプション値 MSS を操作します。

次のシナリオでは、ルータ A およびルータ B は同じ管理ドメイン内にあります。 ルータ C はアクセスできず、ICMP をブロックするので、PMTUD が失敗します。 この状況の回避策として、フラグメンテーションを許可するため、ルータ B で両方向の DF ビットをクリアします。 これはポリシー ルーティングで実行できます。 DF ビットをクリアするための構文は、Cisco IOS® ソフトウェア リリース 12.1(6) 以降で利用可能です。

```
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
```



もう 1 つのオプションは、ルータを通過する SYN パケット上の TCP MSS オプションの値を変更することです（Cisco IOS 12.2(4)T 以降で使用可能）。 これにより、TCP SYN パケット内の MSS オプションの値が低下し、**ip tcp adjust-mss** コマンドでの値（1460）より小さくなります。 その結果、TCP の送信側では、この値に収まる大きさのセグメントを送信します。 IP パケットのサイズは、TCP ヘッダー（20 バイト）と IP ヘッダー（20 バイト）を加えるため、MSS 値（1460 バイト）より 40 バイト大きくなります（1500）。

ip tcp adjust-mss コマンドにより、TCP SYN パケットの MSS を調整できます。 次の構文では、TCP セグメント上の MSS 値が 1460 に低下させられます。 このコマンドは、インターフェイス serial0 での着信と発信両方のトラフィックに影響します。

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

IP トンネルがより広く普及するようになったのに従い、IP フラグメンテーションの問題がさらにまん延するようになりました。 トンネルがフラグメンテーションをさらに発生させている理由は、トンネルのカプセル化による「オーバーヘッド」がパケット サイズに付加されているからです。 たとえば、Generic Routing Encapsulation（GRE）の付加により、24 バイトがパケットに追加されます。そして、この増加によって、パケットが発信側 MTU より大きくなり、フラグメント化が必要になる場合があります。 このドキュメントの後のセクションで、トンネルと IP フラグメンテーションにより発生する種類の問題を、例を挙げて説明しています。

PMTUD が必要とされる一般的なネットワーク トポロジ

PMTUD は、中継リンクの MTU がエンド リンクの MTU より小さいようなネットワーク状況において必要となります。 これらのより小さな MTU リンクが存在する一般的な理由としては、次のものがあります。

- トークン リング（または FDDI）に接続されたエンド ホストで、中間にイーサネット接続がある場合。 これらの両端でのトークン リング（または FDDI）MTU は、中間にあるイーサネット MTU より大きくなります。
- （ADSL でよく使用される）PPPoE では、そのヘッダーに 8 バイトが必要です。 これにより、イーサネットでの有効 MTU が 1492（1500 - 8）に低下します。

GRE、IPsec、および L2TP などのトンネリング プロトコルでも、それぞれのヘッダーとトレーラのための領域が必要です。 これもまた、発信インターフェイスの有効 MTU を低下させます。

次のセクションでは、2 つのエンド ホスト間のいずれかの場所でトンネリング プロトコルが使用される場合の PMTUD の影響を検討します。 前述の 3 つの状況では、この状況が最も複雑であり

、他の状況で発生するすべての問題も含まれます。

トンネルの概要

トンネルとは、トランスポート プロトコル内で、パッセンジャ パケットをカプセル化する方法を提供する、Cisco ルータ上の論理インターフェイスです。 これは、ポイントツーポイントのカプセル化スキームを実装するサービスを提供する設計になっているアーキテクチャです。 トンネリングには、次の 3 つの主要コンポーネントが用意されます。

- パッセンジャ プロトコル (AppleTalk、Banyan VINES、CLNS、DECnet、IP、または IPX)
- キャリア プロトコル：次のいずれかのカプセル化プロトコル。GRE： シスコのマルチプロトコル キャリア プロトコル。 インターネット上での PMTUD の不具合についての詳細は、RFC 2784 を および RFC 1701 問い合わせてくださいIP トンネル内の IP：詳細は、『RFC 2003』を参照してください。 問い合わせてください
- トランスポート プロトコル：カプセル化プロトコルを伝送するために使用されるプロトコル

このセクションのパケットは、IP トンネリングの概念を図示しています。この場合、GRE がカプセル化プロトコル、IP がトランスポート プロトコルです。 また、パッセンジャ プロトコルも IP です。 この場合、IP はトランスポート プロトコルおよびパッセンジャ プロトコル両方です。

ノーマル パケット

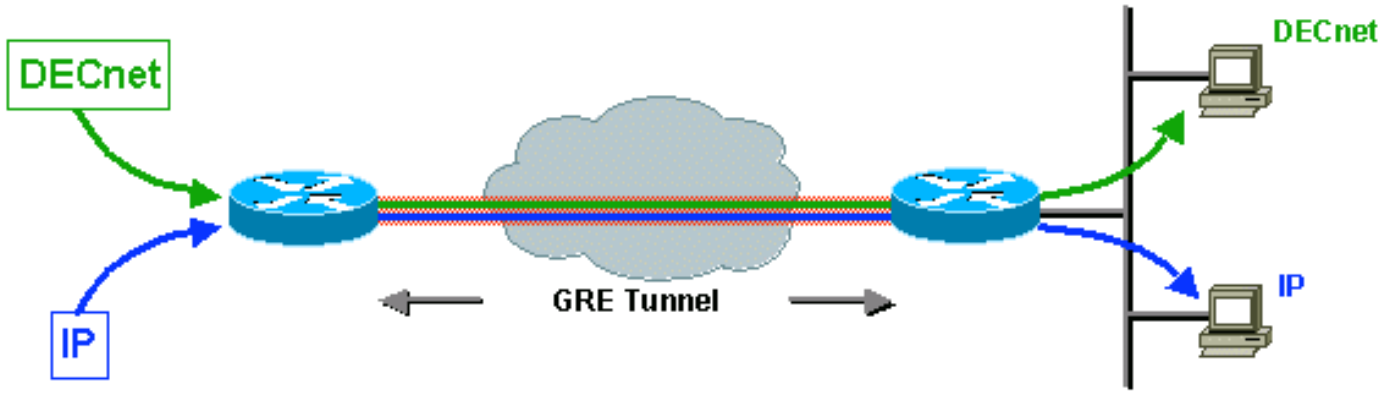
IP TCP Telnet

トンネル パケット

IP GRE IP TCP Telnet

- IP はトランスポート プロトコルです。
- GRE はカプセル化プロトコルです。
- IP はパッセンジャ プロトコルです。

次の例では、キャリアとして GRE を使った、パッセンジャ プロトコルとしての IP および DECnet のカプセル化を示しています。 これは、キャリア プロトコルによって複数のパッセンジャ プロトコルのカプセル化が可能であることを示しています。



IP バックボーンによって隔てられた 2 つの非隣接の非 IP ネットワークが存在する状況では、ネットワーク管理者はトンネリングを検討することができます。 この非隣接ネットワークで DECnet が実行されている場合、バックボーンに DECnet を設定することにより、これらを接続することを、管理者が望まない場合があります。 管理者は、IP ネットワークのパフォーマンスに影響を及ぼす可能性があるため、バックボーン帯域幅を消費する DECnet ルーティングを許可したくない場合があります。

実行可能な代案は、IP バックボーンを介して DECnet をトンネル化することです。 トンネリングによって IP 内部に DECnet パケットがカプセル化され、バックボーンを介してカプセル化を外すントンネル エンドポイントに送信されます。また、DECnet を介した DECnet パケットの宛先へのルーティングが可能になります。

他のプロトコル内でのトラフィックのカプセル化により、次の利点が実現されます。

- エンドポイントではプライベート アドレスが使用され（RFC 1918 ）、バックボーンではこれらのアドレスのルーティングはサポートされない。
- WAN またはインターネット介した Virtual Private Network（VPN; バーチャル プライベート ネットワーク）を可能にする。
- 単一プロトコルのバックボーンを介して、非隣接マルチプロトコル ネットワークを統合する。
- バックボーンまたはインターネット上のトラフィックを暗号化する。

このドキュメントの残りの部分では、IP はパッセンジャ プロトコルとして、また IP はトランスポート プロトコルとして使用します。

トンネル インターフェイスに関する考察

トンネリングを実行する場合の注意事項は次のとおりです。

- Cisco IOS リリース 11.1 で、GRE トンネルのファースト スイッチングが導入されました。また、バージョン 12.0 では、CEF スイッチングが導入されています。 マルチポイント GRE トンネルの CEF スイッチングは、バージョン 12.2(8)T で導入されています。 トンネル エンドポイントでのカプセル化とカプセル化解除は、プロセス スイッチングだけがサポートされていた Cisco IOS の初期バージョンでは処理が低速でした。
- パケットをトンネリングする場合には、セキュリティおよびトポロジの問題があります。 トンネルは、Access Control List（ACL; アクセス コントロール リスト）およびファイアウォールをバイパスできます。 ファイアウォールを介してトンネル化する場合、トンネリングしているパッセンジャ プロトコルの種類にかかわらず、基本的にはファイアウォールをバイパスします。 したがって、パッセンジャ プロトコルで任意のポリシーを実施するために、トンネルのエンドポイントにファイアウォール機能を備えることが推奨されます。
- トンネリングでは、遅延の増大により、タイマーで制限されたトランスポート プロトコル（たとえば DECnet）に問題が発生する場合があります。
- 異なる速度のリンクが含まれた環境（高速 FDDI リングと低速 9600 bps 電話回線など）にわたるトンネリングでは、パケットの順序が入れ替わる問題が発生する場合があります。 混合メディア ネットワークでは、一部のパッセンジャ プロトコルの機能は不完全です。
- ポイントツーポイント トンネルは、物理リンク上の帯域幅を使い果たす可能性があります。ルーティング プロトコルを複数のポイントツーポイント トンネルを介して実行する場合、各トンネル インターフェイスには帯域幅が割り当てられている点、およびトンネルが実行されている物理インターフェイスにも帯域幅が割り当てられている点に注意してください。 たとえば、10 MB リンクを介して 100 のトンネルが実行されている場合、トンネル帯域幅を 100 KB に設定します。 トンネルのデフォルト帯域幅は、9 KB です。
- ルーティング プロトコルで、「実」リンクを介したトンネルが優先される場合があります。この理由は、トンネルが、最も低いコスト パスを持つ 1 ホップ リンクであるように誤って認識されるためです。ところが、実際はトンネルにはそれ以上のホップが含まれ、他のパスよりも非常にコスト高です。 この問題は、ルーティング プロトコルの適切な設定により軽減できます。 物理インターフェイス上で実行されているルーティング プロトコルとは異なるルー

ティング プロトコルを、トンネル インターフェイスで実行することを検討する必要がある場合があります。
- 再帰ルーティングの問題は、トンネル宛先への適切なスタティック ルートを設定することにより回避できます。 再帰ルートとは、「トンネル宛先」への最善パスがトンネル自体を通っている場合を指します。 この状況では、トンネル インターフェイスが不安定になります。 再帰ルーティングの問題が発生している場合、次の工ラーが表示されます。%TUN-RECURDOWN
```
Interface Tunnel 0
temporarily disabled due to recursive routing
```

トンネルのエンドポイントにおいて PMTUD に関与するルータ

トンネルのエンドポイントとなっているルータには 2 つの異なる PMTUD の役割があります。

- 1 番目の役割では、ルータはホスト パケットを転送します。 PMTUD 処理のために、ルータは元のデータ パケットの DF ビットおよびパケット サイズを確認し、必要に応じて適切な処理を行う必要があります。
- 2 番目の役割は、ルータがトンネル パケット内に元の IP パケットをカプセル化した後、実行されます。 この段階では、ルータは PMTUD およびトンネル IP パケットに関して、ホストのような動作をします。

PMTUD に関して、ホスト IP パケットを転送する 1 番目の役割でルータが動作する場合の状況から解説します。 この役割は、ルータがトンネル パケット内にホスト IP パケットをカプセル化する前に実行されます。

ルータがホスト パケットの転送側となる場合、次のアクションが実行されます。

- DF ビットが設定されているかどうかの確認。
- トンネルが対応できるパケット サイズの確認。
- フラグメント化（パケットが大きすぎて DF ビットが設定されていない場合）、フラグメントのカプセル化、および送信。 または
- パケットの廃棄（パケットが大きすぎて DF ビットが設定されている場合）および送信側への ICMP メッセージの送信。
- カプセル化（パケットが大きすぎない場合）および送信。

一般的に、カプセル化後のフラグメント化（2 つのカプセル化フラグメントの送信）、またはフラグメント化後のカプセル化（2 つのカプセル化フラグメントの送信）のどちらかを選択できます。

このセグメントでは、IP パケット カプセル化とフラグメント化の仕組みを説明するいくつかの例、および例として挙げられたネットワークを通過するパケットと PMTUD とのインタラクションを示す 2 つのシナリオについて、詳細に説明します。

次の 1 番目の例では、（トンネル発信元の）ルータが転送ルータの役割を果たす場合のパケットの状態を示します。 PMTUD 処理のために、ルータは元のデータ パケットの DF ビットおよびパケット サイズを確認し、適切な処理を行う必要があることを思い出してください。 この例では、トンネルの GRE カプセル化を使用しています。 確認できるように、GRE はカプセル化の前にフラグメント化を実行します。 後で挙げる例では、カプセル化の後にフラグメント化が実行されるシナリオを説明します。

例 1 では DF ビットが設定されておらず（DF = 0）、GRE トンネル IP MTU は 1476（1500 - 24）です。

例 1

1. （トンネル発信元の）転送ルータは、送信側ホストから、DF ビットがクリアされた（DF = 0）1500 バイトのデータグラムを受信します。 このデータグラムは、20 バイトの IP ヘッダーと 1480 バイトの TCP ペイロードから構成されています。
2. GRE オーバーヘッド（24 バイト）が追加されると、パケットは IP MTU に対して大きくなりすぎるため、フォワーディング ルータによってデータグラムが 1476 バイト（20 バイトの IP ヘッダー + 1456 バイトの IP ペイロード）および 44 バイト（20 バイトの IP ヘッダー + 24 バイトの IP ペイロード）の 2 つのフラグメントに分割されます。これにより、GRE カプセル化による追加があっても、パケットは発信物理インターフェイス MTU より大きくはなりません。
3. フォワーディング ルータでは、GRE カプセル化による追加があります。元の IP データグラムの各フラグメントに対して、4 バイトの GRE ヘッダーと 20 バイトの IP ヘッダーが追加分になります。 これにより、これら 2 つの IP データグラムは 1500 バイトおよび 68 バイトの長さとなります。これらのデータグラムはフラグメントとしてではなく、個々の IP データグラムとして認識されます。
4. トンネルの宛先側ルータでは、GRE カプセル化による付加分が元のデータグラムの各フラグメントから削除され、1476 バイトと 24 バイトの長さの 2 つの IP フラグメントが残されます。 これらの IP データグラム フラグメントは、このルータによって受信側のホストに別々に転送されます。
5. 受信側ホストは、これら 2 つのフラグメントを元のデータグラムに再構成します。

シナリオ 5 では、ネットワーク トポロジの観点での転送ルータの役割を図示します。

次の例では、ルータは転送ルータと同じ役割を果たしますが、この場合は DF ビットが設定されています（DF = 1）。

例 2

1. トンネル発信元の転送ルータは、送信側ホストから 1500 バイトのデータグラム（DF = 1）を受信します。
2. DF ビットが設定され、データグラム サイズ（1500 バイト）が GRE トンネル IP MTU（1476）より大きいので、ルータはデータグラムを廃棄し、「ICMP Fragmentation Needed but DF set」メッセージをデータグラムの発信元に送信します。 ICMP メッセージによって、MTU が 1476 であることが送信側に警告されます。
3. 送信側ホストは ICMP メッセージを受け取り、元のデータを送信する際に 1476 バイトの IP データグラムを使用します。
4. この IP データグラムの長さ（1476 バイト）は、今回は GRE トンネル IP MTU の値に等しいため、ルータはこの IP データグラムに GRE カプセル化を行います。
5. （トンネル宛先側の）受信側ルータは、IP データグラムの GRE カプセル化による付加分を削除してから、それを受信側ホストに送信します。

次に、PMTUD およびトンネル IP パケットに関して、ルータが送信側ホストとして 2 番目の役割を果たす場合の状況を説明します。 この役割は、ルータがトンネル パケット内に元の IP パケットをカプセル化した後で実行されることを思い出してください。

> **注:** デフォルトでは、ルータは生成する GRE トンネル パケットに対して PMTUD を実行しません。 tunnel path-mtu-discovery コマンドを使用して、GRE-IP トンネル パケットに対して PMTUD を有効にできます。

例 3 では、GRE トンネル インターフェイス上の IP MTU に収まるほど小さい IP データグラムをホストが送信する場合の状況を示します。 この場合、DF ビットを設定またはクリア (1 または 0) することが可能です。 この GRE トンネル インターフェイスでは tunnel path-mtu-discovery コマンドが設定されていないので、ルータによる GRE-IP パケットへの PMTUD は実行されません。

例 3

1. トンネル発信元の転送ルータは、送信側ホストから 1476 バイトのデータグラムを受信します。
2. このルータでは GRE 内で 1476 バイトの IP データグラムがカプセル化されて、1500 バイトの GRE IP データグラムが作成されます。 GRE IP ヘッダー内の DF ビットはクリアされます (DF = 0)。 次に、このルータはこのパケットをトンネルの宛先に転送します。
3. トンネルの発信元と宛先の間に、1400 のリンク MTU を持つルータが存在すると仮定します。 DF ビットがクリアされている (DF = 0) ので、このルータではトンネル パケットがフラグメント化されます。 この例では、最も外側の IP がフラグメント化されるので、GRE、内側の IP、および TCP ヘッダーは最初のフラグメントだけに表示されていることを憶えておいてください。
4. このトンネルの宛先ルータで、GRE トンネル パケットを再構成する必要があります。
5. GRE トンネル パケットが再構成されると、ルータは GRE IP ヘッダーを削除し、元の IP データグラムをその宛先へ送信します。

次の例では、PMTUD およびトンネル IP パケットに関して、ルータが送信側ホストの役割を果たす場合の状況を説明します。 この場合、元の IP ヘッダー内で DF ビットがセットされ (DF = 1)、tunnel path-mtu-discovery コマンドが設定されているので、内側の IP ヘッダーから外側の (GRE + IP) ヘッダーに DF ビットがコピーされます。

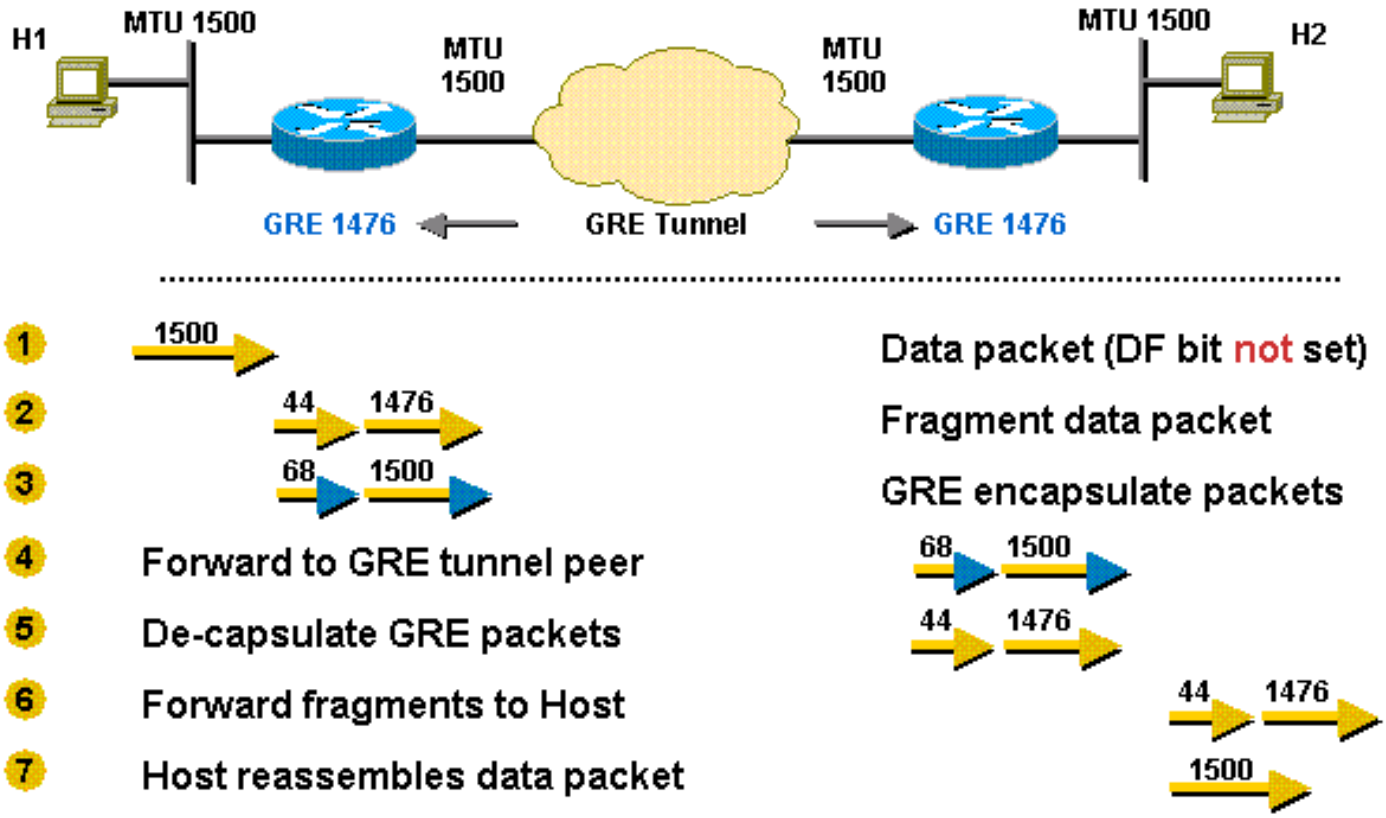
例 4

1. トンネル発信元の転送ルータは、送信側ホストから 1476 バイトのデータグラム (DF = 1) を受信します。
2. このルータでは GRE 内で 1476 バイトの IP データグラムがカプセル化されて、1500 バイトの GRE IP データグラムが作成されます。 元の IP データグラムの DF ビットが設定されているため、この GRE IP ヘッダーでは DF ビットが設定されます (DF = 1)。 次に、このルータはこのパケットをトンネルの宛先に転送します。
3. 再度、トンネルの発信元と宛先の間に、1400 のリンク MTU を持つルータが存在すると仮定します。 DF ビットが設定されている (DF = 1) ので、このルータではトンネル パケットのフラグメント化が行われません。 このルータはパケットを廃棄し、ICMP エラー メッセージをトンネルの発信元ルータに送信する必要があります。この理由は、これがパケット上の発信元 IP アドレスであるからです。
4. トンネル発信元の転送ルータは、この ICMP エラー メッセージを受信し、GRE トンネル IP MTU を 1376 (1400 - 24) に低下させます。 送信側ホストが次にデータを 1476 バイトの IP パケットで再送信すると、このパケットは大きすぎることになるため、このルータは 1376 の MTU 値を付けて ICMP エラー メッセージを送信側に送ります。 送信側ホストがデータを再送信する際には、1376 バイトの IP パケットで送信し、このパケットは GRE トンネルを介して受信側ホストに到着します。

シナリオ 5

このシナリオでは、GRE フラグメンテーションを解説します。 GRE のカプセル化の前にフラグ

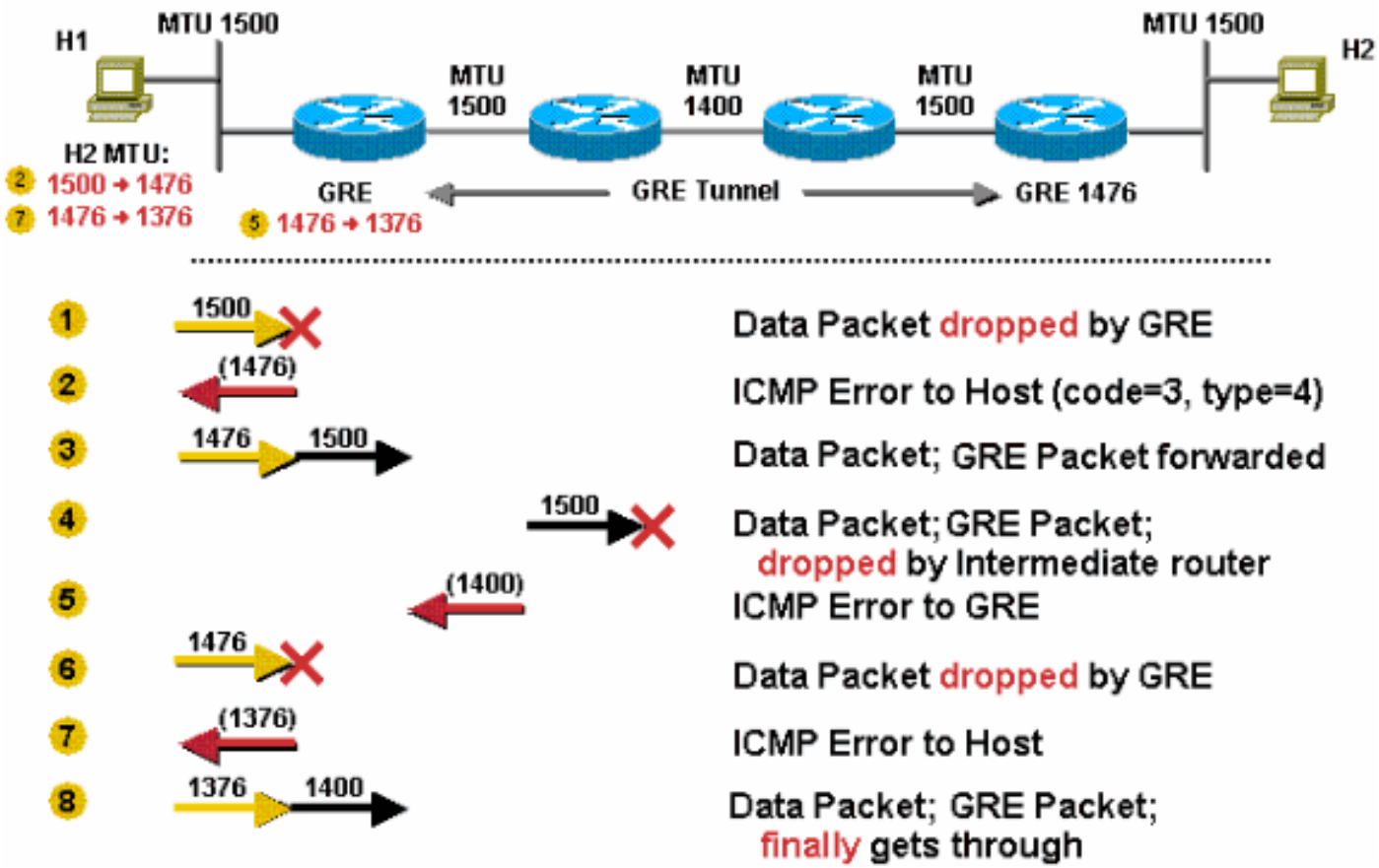
メント化し、次にデータ パケットの PMTUD を実行することと、IP パケットが GRE によってカプセル化される場合には、DF ビットがコピーされないことを憶えておいてください。 このシナリオでは、DF ビットは設定されていません。 GRE トンネル インターフェイスの IP MTU は、デフォルトでは物理インターフェイスの IP MTU より 24 バイト少ないので、GRE インターフェイスの IP MTU は 1476 となります。



1. 送信側は 1500 バイトのパケット (20 バイト IP ヘッダー + 1480 バイトの TCP ペイロード) を送信します。
2. GRE トンネルの MTU は 1476 なので、1500 バイトのパケットは 1476 バイトと 44 バイトの 2 つの IP フラグメントに分割されます。それぞれの IP フラグメントには、24 バイトの GRE ヘッダーが付加されることを見込んでいます。
3. 24 バイトの GRE ヘッダーが各 IP フラグメントに付加されます。 したがって、フラグメントはそれぞれ 1500 バイト (1476 + 24) および 68 バイト (44 + 24) になります。
4. 2 つの IP フラグメントを含む GRE + IP パケットが、GRE トンネル ピア ルータに転送されます。
5. GRE トンネル ピア ルータが、2 つのパケットから GRE ヘッダーを削除します。
6. このルータは、2 つのパケットを宛先ホストに転送します。
7. 宛先ホストは、この IP フラグメントを元の IP データグラムに再構成します。

シナリオ 6

このシナリオはシナリオ 5 に類似していますが、今回は DF ビットが設定されています。 シナリオ 6 では、ルータは GRE + IP トンネル パケットに PMTUD を実行するように、**tunnel path-mtu-discovery** コマンドで設定されています。また、DF ビットは元の IP ヘッダーから GRE IP ヘッダーにコピーされます。 ルータでは、GRE + IP パケットの ICMP エラーを受信すると、GRE トンネル インターフェイス上の IP MTU を低下させます。 再度、デフォルトでは GRE トンネル IP MTU が物理インターフェイス MTU より 24 バイト少なく設定されているので、GRE IP MTU が 1476 であることを思い出してください。 さらに、GRE トンネル パス内に 1400 の MTU リンクが存在することにも注意してください。



1. ルータは、1500 バイトのパケット (20 バイトの IP ヘッダー + 1480 バイトの TCP ペイロード) を受信し、このパケットを廃棄します。 ルータがこのパケットを廃棄する理由は、これが GRE トンネル インターフェイス上の IP MTU (1476) よりも大きいためです。
2. ルータは、ネクスト ホップの MTU が 1476 であることを通知する ICMP エラーを送信側に送ります。 ホストは、通常は、その宛先のホスト ルートとして、この情報をルーティング テーブル内に記録します。
3. 送信側ホストは、データを再送信する際に 1476 バイトのパケット サイズを使用します。 GRE ルータでは、24 バイトの GRE カプセル化付加分を追加し、1500 バイトのパケットを送り出します。
4. 1500 バイトのパケットは 1400 バイトのリンクを通過できないので、中継ルータによって廃棄されます。
5. 中継ルータは、1400 のネクスト ホップ MTU 値を付けて、ICMP (タイプ 3、コード 4) を GRE ルータに送信します。 GRE ルータでは、これを 1376 (1400 - 24) に低下させて、GRE インターフェイスでの内部 IP MTU 値を設定します。 この変更は、**debug tunnel** コマンドを使用するときのみ確認できます。 これは **show ip interface tunnel<#>** コマンドの出力では確認できません。
6. ホストが次に 1476 バイトのパケットを再送信する場合、このパケットは GRE トンネル インターフェイスの現在の IP MTU (1376) より大きいので、GRE ルータはこれを廃棄します。
7. GRE ルータは、1376 のネクスト ホップ MTU 値を付けて、別の ICMP (タイプ 3、コード 4) を送信側に送り、ホストでは新しい値で現在の情報を更新します。
8. ホストは再度データを再送信しますが、今回はより小さい 1376 バイトのパケットで送信します。GRE はカプセル化の 24 バイトを追加し、これを転送します。 今回は、パケットは GRE トンネル ピアに到着し、ここでカプセル化解除され、宛先ホストに送信されます。**注:** このシナリオにおいて、転送ルータ上で **tunnel path-mtu-discovery** コマンドが設定されておらず、さらに GRE トンネルを介して転送されたパケット内で DF ビットが設定されている

場合、Host 1 は TCP/IP パケットを Host 2 に送信できますが、これらのパケットは途中、1400 MTU リンクでフラグメント化されます。 さらに、GRE トンネル ピアでは、これらをカプセル化解除して転送する前に、再構成する必要があります。

「ピュア」IPsec トンネル モード

IP セキュリティ（IPsec）プロトコルは、IP ネットワークを介して送信される情報にプライバシ、整合性、および信憑性を提供する、標準ベースの方式です。 IPsec では、IP ネットワーク層での暗号化が提供されます。 IPsec では、少なくとも 1 つの IP ヘッダー（トンネル モード）が追加されるので、IP パケットが長くなります。 付加されたヘッダは依存 IPSec構成 モードの長さが異なりますが、~58 バイトを超過しません（Encapsulating Security Payload （ESP）および ESP 認証（ESPauth））を超えることはありません。

IPsec には、トンネル モードおよびトランスポート モードの 2 つのモードがあります。

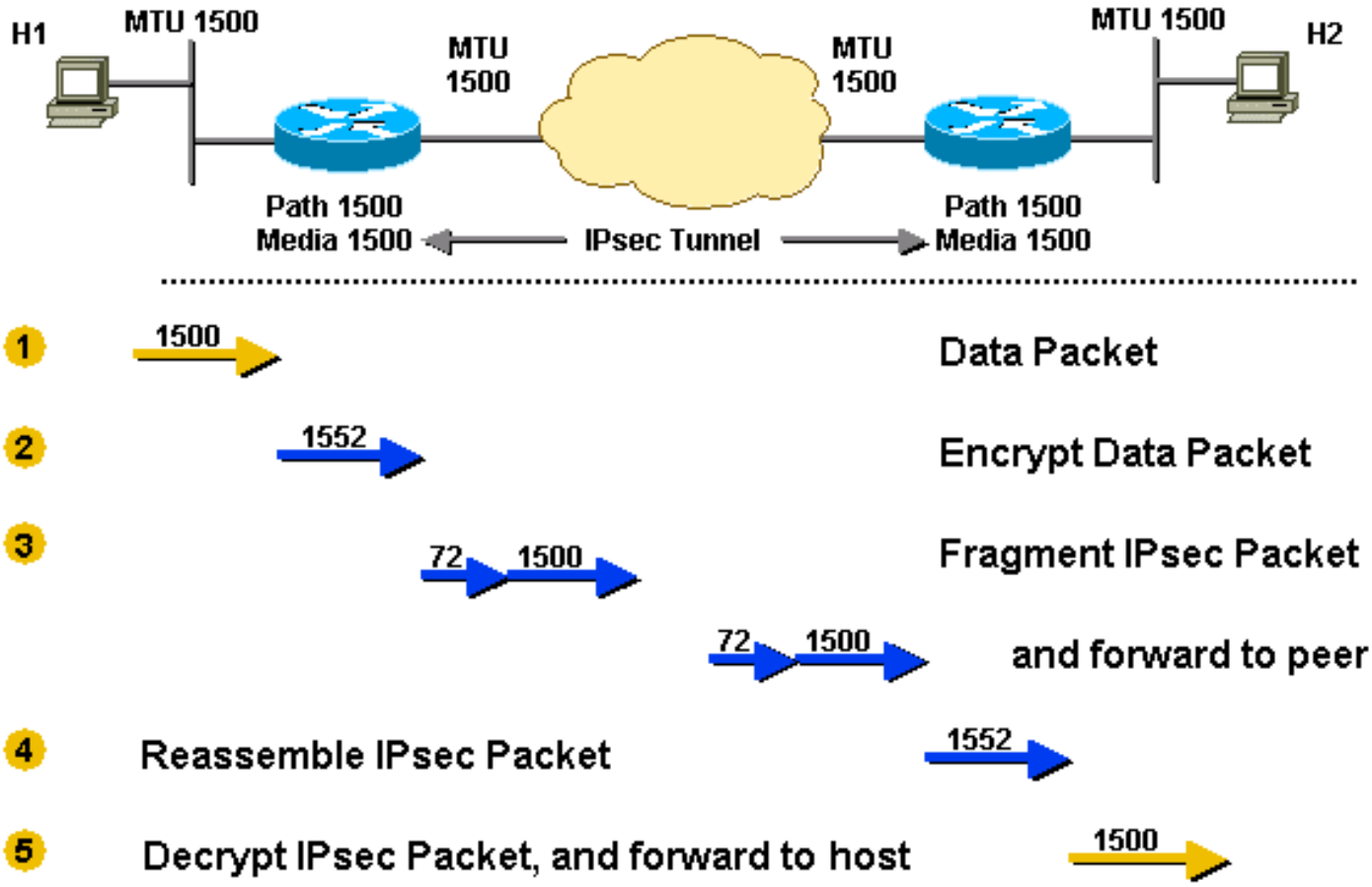
- トンネル モードがデフォルトのモードです。 トンネル モードでは、元の IP パケットはすべて保護され（暗号化、認証、またはその両方）、IPsec のヘッダーとトレーラでカプセル化されます。 次に、新規の IP ヘッダーがパケットの先頭に付加されて、発信元および宛先に IPsec エンドポイント（ピア）が指定されます。 トンネル モードは任意のユニキャスト IP トラフィックで使用でき、IPsec がホストからのトラフィックを IPsec ピアの後方で保護している場合に使用する必要があります。 たとえば、トンネル モードは VPN で使用されます。この場合、保護されたあるネットワーク上のホストが保護された別のネットワーク上のホストにパケットを送信するのに、一対の IPsec ピアが経由されます。 VPN では、IPsec「トンネル」は IPsec ピア ルータ間のこのトラフィックを暗号化することにより、ホスト間の IP トラフィックを保護します。
- （トランスフォーム定義で、サブコマンド **mode transport** で設定される）トランスポート モードでは、元の IP パケットのペイロードだけが保護（暗号化、認証、またはその両方）されます。 ペイロードは、IPsec のヘッダーとトレーラでカプセル化されます。 IP プロトコル フィールドの ESP（50）への変更を除き、元の IP ヘッダーはそのままの状態です。また、元のプロトコル値は、パケットが復号化される際の復元のために IPsec トレーラに保存されます。 トランスポート モードは、IP トラフィックが IPsec ピア自体の間で保護される場合にだけ使用され、パケット上の発信元および宛先の IP アドレスは、IPsec ピア アドレスと同じになります。 IPsec トランスポート モードが使用されるのは、通常、最初の IP データ パケットをカプセル化に別のトンネリング プロトコル（GRE など）が使用され、次に IPsec により GRE トンネル パケットを保護するような場合だけです。

IPsec では常に、データ パケットおよび IPsec 自体のパケットのために PMTUD が実行されます。 IPsec IP パケットの PMTUD 処理を変更するために、IPsec 設定コマンドがあります。IPsec では DF ビットに関して、クリア、設定、またはデータ パケットの IP ヘッダーから IPsec の IP ヘッダーへのコピーが可能です。 これは「DF ビット上書き機能」と呼ばれます。

注: IPsec でハードウェアでの暗号化を行う場合、カプセル化の後のフラグメンテーションを必ず回避する必要があります。 ハードウェアでの暗号化では、ハードウェアによっては約 50 Mbps のスループットが提供されますが、IPsec パケットがフラグメント化される場合、このスループットの 50 ~ 90 % が失われます。 フラグメント化された IPsec パケットが再構成のためにプロセス交換された後、復号化のためにハードウェア暗号化エンジンにわたされることがこの損失の原因です。 スループットのこの損失によって、ハードウェア暗号化スループットがソフトウェア暗号化のパフォーマンス レベル（2 ~ 10 Mbps）にまで低下する可能性があります。

シナリオ 7

このシナリオでは、実行中の IPsec フラグメンテーションを図示します。 このシナリオでは、全パスでの MTU は 1500 です。 このシナリオでは、DF ビットは設定されていません。

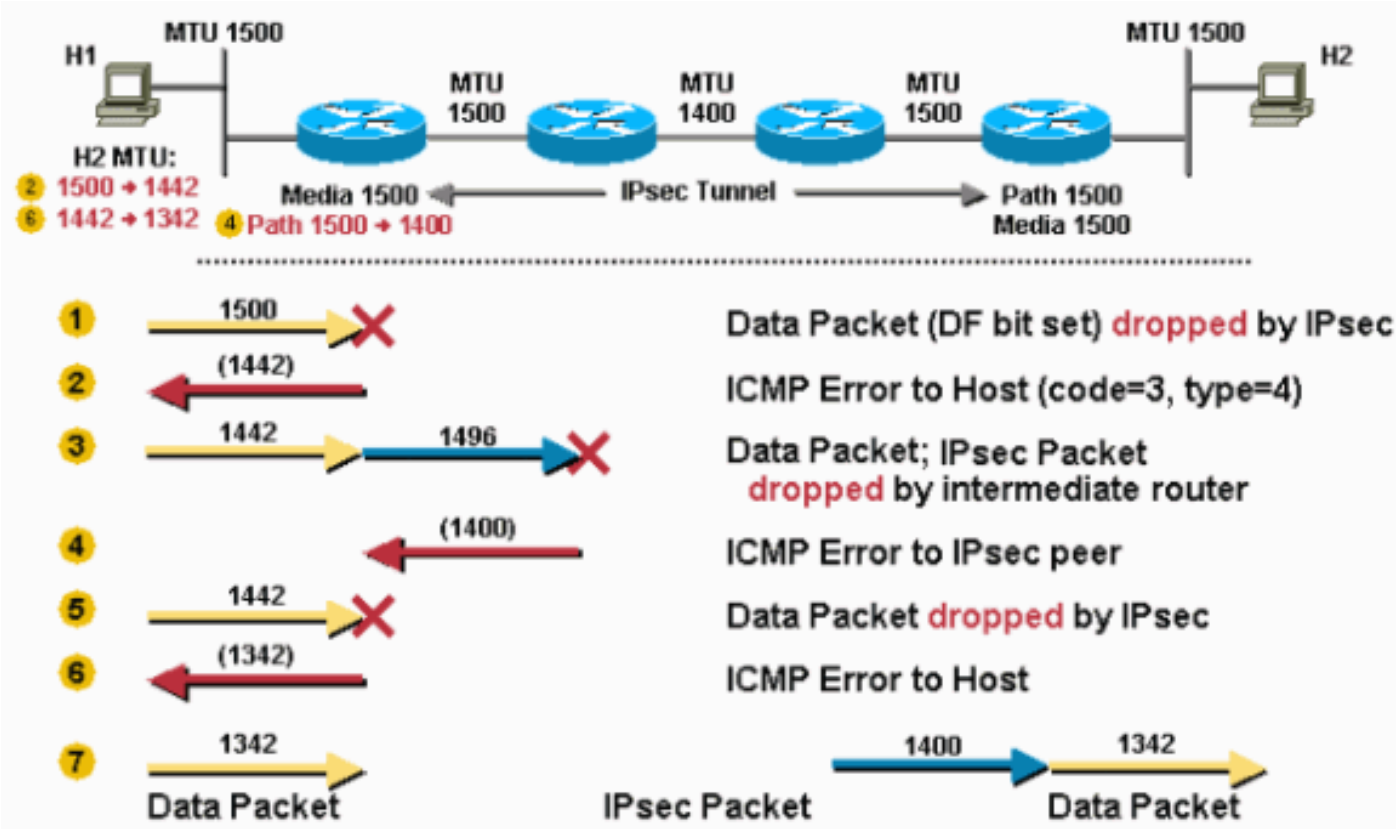


1. ルータは、Host 2 を宛先とした 1500 バイトのパケット (20 バイト IP ヘッダー + 1480 バイトの TCP ペイロード) を受信します。
2. 1500 バイトのパケットが IPsec によって暗号化され、52 バイトのオーバーヘッド (IPsec ヘッダー、トレーラ、および追加の IP ヘッダー) が追加されます。 これにより、IPsec は 1552 バイトのパケットを送信する必要があります。 送信 MTU が 1500 なので、このパケットをフラグメント化する必要があります。
3. この IPsec パケットから 2 つのフラグメントが作成されます。 フラグメンテーション中、2 番目のフラグメントに対してさらに 20 バイトの IP ヘッダーが追加され、結果として 1500 バイトのフラグメントと 72 バイトの IP フラグメントになります。
4. IPsec トンネル ピア ルータはこれらのフラグメントを受信し、追加の IP ヘッダーを取り除き、さらに、これらの IP フラグメントを結合して元の IPsec パケットに戻します。 次に、IPsec によってこのパケットが復号化されます。
5. ルータは次に、元の 1500 バイトのデータ パケットを Host 2 に転送します。

シナリオ 8

このシナリオはシナリオ 6 に類似しています。ただし、この場合は元のデータ パケットに DF ビットが設定されており、IPsec トンネル ピア間のパスにその他のリンクより低い MTU を持つリンクが存在する点が異なります。 このシナリオでは、「トンネルのエンドポイントにおいて PMTUD に関与するルータ」のセクションで説明したような、両方の PMTUD の役割を果たす IPsec ピア ルータの動作を説明します。

このシナリオでは、フラグメンテーションに必要となる、IPsec PMTU の低い値への変更の状況を示します。 IPsec がパケットを暗号化する時に、DF ビットが内側の IP ヘッダーから外側の IP ヘッダーにコピーされることに注意してください。 メディア MTU および PMTU の値は、IPsec Security Association（SA; セキュリティ結合）内に格納されます。 メディア MTU は、発信ルータ インターフェイスの MTU に基づいています。また、PMTU は、IPsec ピア間のパスで発生する最小 MTU に基づいています。 IPsec では、フラグメント化の前に、パケットがカプセル化/暗号化されることに注意してください。



1. ルータは、1500 バイトのパケットを受信して廃棄します。この理由は、IPsec オーバーヘッドが追加される場合、パケットが PMTU（1500）より大きくなるからです。
2. ルータは、ネクスト ホップの MTU が 1442（1500 - 58 = 1442）であることを通知する、ICMP メッセージを Host 1 に送ります。 この 58 バイトとは、IPsec ESP および ESPauth を使用する場合の最大 IPsec オーバーヘッドです。 実際の IPsec オーバーヘッドは、この値より最大 7 バイト小さい値となる場合があります。 Host 1 は通常、宛先（Host 2）のホスト ルートとして、ルーティング テーブル内にこの情報を記録します。
3. Host 1 は、Host 2 に対する PMTU を 1442 に低下させるので、データを Host 2 に再送信する場合により小さい（1442 バイト）パケットを送信します。 ルータは 1442 バイトのパケットを受信し、IPsec は 52 バイトの暗号化オーバーヘッドを追加するので、結果として IPsec のパケットは 1496 バイトとなります。 このパケットは、ヘッダー内に DF ビットが設定されているので、MTU リンクが 1400 バイトの中間ルータによって廃棄されます。
4. パケットを廃棄するこの中間ルータは、ICMP メッセージを IPsec パケットの送信側（1 番目のルータ）に送り、ネクスト ホップの MTU が 1400 バイトであることを伝えます。 この値は、IPsec SA PMTU 内に記録されます。
5. 次に Host 1 が 1442 バイトのパケットを再送信する（この確認応答は受信していません）と、IPsec はパケットを廃棄します。 ルータはパケットを再度廃棄します。この理由は、IPsec オーバーヘッドがパケットに追加される場合、パケットが PMTU（1400）より大きくなるからです。
6. ルータは、ネクスト ホップの MTU が 1342（1400 - 58 = 1342）であることを通知する、

ICMP メッセージを Host 1 に送ります。 Host 1 は、この情報を再度記録します。
7. Host 1 がデータを再送信する場合、より小さなサイズのパケット (1342) を使用します。 このパケットはフラグメンテーションを必要とせず、IPsec トンネルを介して Host 2 に到達します。

GRE と IPsec の統合

GRE トンネルの暗号化に IPsec が使用される場合、フラグメンテーションと PMTUD のより複雑なインタラクションが発生します。 IPsec では IP マルチキャスト パケットがサポートされていないため、IPsec と GRE が次の方法で組み合されることになります。これは、IPsec VPN ネットワークではダイナミック ルーティング プロトコルを実行できないことを意味します。 GRE トンネルはマルチキャストをサポートしているので、まず GRE トンネルを使用して、GRE IP ユニキャスト パケット内のダイナミック ルーティング プロトコル マルチキャスト パケットをカプセル化できます。次に、これを IPsec により暗号化できます。 これを実行すると、多くの場合、GRE に加えて IPsec がトランスポート モードで展開されます。この理由は、IPsec ピアと GRE トンネルのエンドポイント (ルータ) は同じものであり、トランスポート モードでは IPsec オーバーヘッドの 20 バイトが節約されるからです。

注目すべき状況の 1 つに、IP パケットが 2 つのフラグメントに分割され、GRE によってカプセル化される場合があります。 この場合、IPsec では 2 つの独立した GRE + IP のパケットに対応することになります。 デフォルト設定では、多くの場合、これらのパケットの 1 つは大きいため、暗号化された後でフラグメント化される必要があります。 IPsec ピアは、復号化の前にこのパケットを再構成する必要があります。 送信側ルータでの、この「2 重のフラグメンテーション」(GRE の前に 1 回、IPsec の後に 1 回) は、遅延を増大させ、スループットを低下させます。また、再構成はプロセス交換されるので、この状態が発生するたびに受信側ルータ上で CPU ヒットが発生します。

この状況は、GRE と IPsec 両方からのオーバーヘッドに対処するほど低く、GRE トンネル インターフェイス上の「ip mtu」を設定することによって回避できます (デフォルトでは、GRE トンネル インターフェイスの「ip mtu」は、実際の送信インターフェイスの MTU である GRE オーバーヘッドのバイトに設定されています)。

次の表では、発信物理インターフェイスの MTU が 1500 であると仮定して、各トンネル/モードの組み合せに推奨される MTU 値を掲載しています。

トンネルの組み合せ	必要な特定の MTU	推奨される MTU
GRE + IPsec (トランスポート モード)	1440 バイト	1400 バイト
GRE + IPsec (トンネル モード)	1420 バイト	1400 バイト

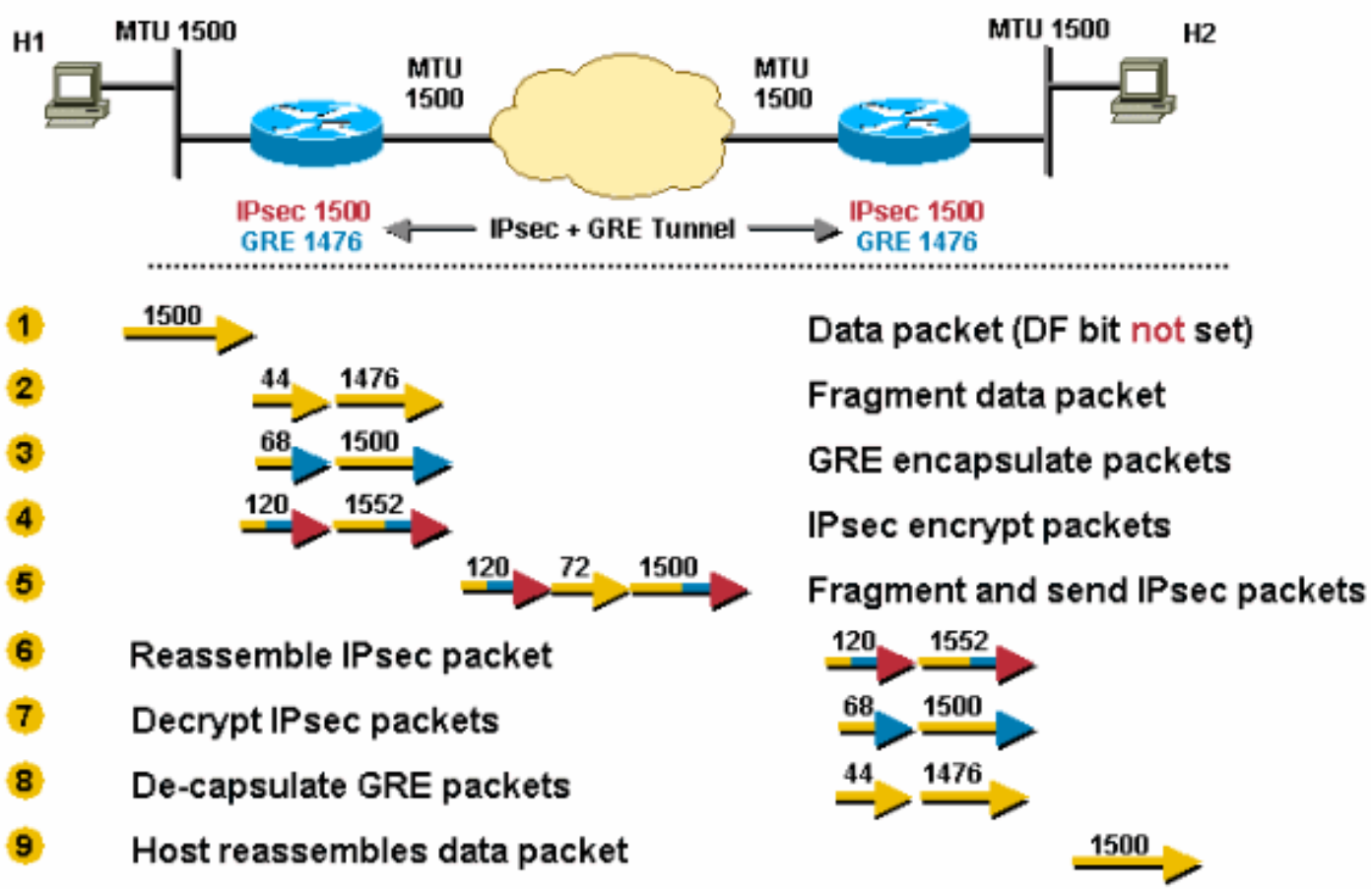
> **注:** 一般的な GRE + IPsec モードの組合せの大部分に対応しているため、MTU 値には 1400 が推奨されます。 また、追加的な 20 バイトまたは 40 バイトのオーバーヘッドを許可することへの認識できるマイナス面はありません。 1 つの値を記憶して設定し、この値を使用してほぼすべてのシナリオに対応するほうが簡単です。

シナリオ 9

IPsec が GRE に加えて展開されています。 発信物理 MTU は 1500、IPsec PMTU は 1500、そして GRE IP MTU は 1476 (1500 - 24 = 1476) です。 このため、TCP/IP パケットは 2 回フラグメント化されます。GRE の前に 1 回と IPsec の後に 1 回です。 パケットは GRE カプセル化の前にフラグメント化され、フラグメント化された GRE パケットの 1 つが IPsec 暗号化の後で再度

フラグメント化されます。

GRE トンネル上で「ip mtu 1440」（IPsec トランスポート モード）または「ip mtu 1420」（IPsec トンネル モード）を設定すると、このシナリオでの 2 重のフラグメンテーションの可能性が解消されます。



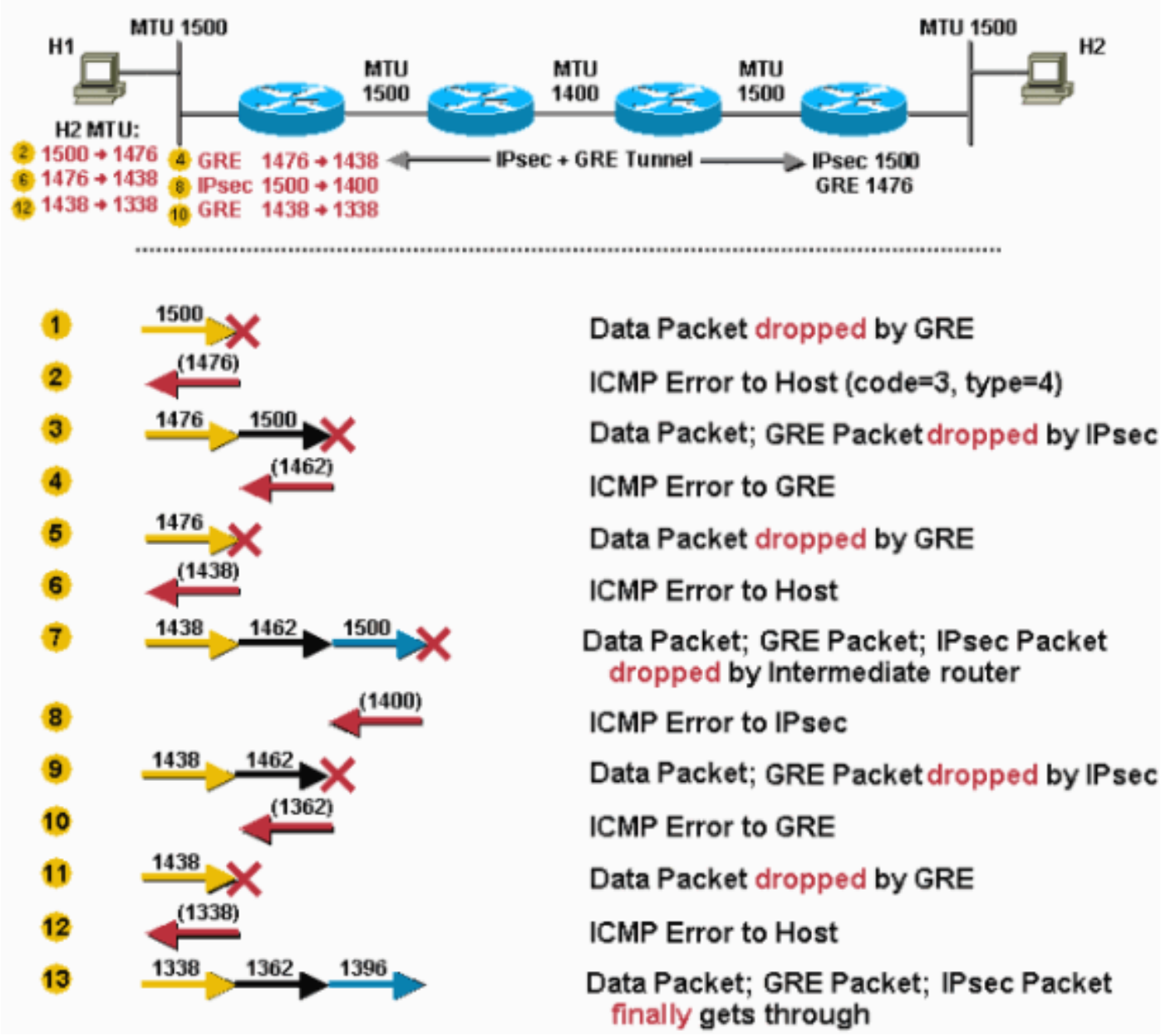
1. ルータは 1500 バイトのデータグラムを受信します。
2. カプセル化の前に、GRE により、1500 バイトのパケットが 1476 バイト（1500 - 24 = 1476）と 44 バイト（24 データ + 20 IP ヘッダー）の 2 つの断片にフラグメント化されます。
3. GRE では IP フラグメントをカプセル化し、各パケットに 24 バイトが追加されます。 この結果として、それぞれ 1500 バイト（1476 + 24 = 1500）および 68 バイト（44 + 24）の、2 つの GRE + IPsec パケットとなります。
4. この 2 つのパケットは IPsec により暗号化され、1552 バイトと 120 バイトのパケットを提供するため、カプセル化オーバーヘッドの 52 バイト（IPsec トンネル モード）がそれぞれに追加されます。
5. 1552 バイトの IPsec パケットは送信 MTU（1500）より大きいため、ルータによりフラグメント化されます。 1552 バイトのパケットは、1500 バイトのパケットと 72 バイトのパケットに分割されます（後者のフラグメントには、52 バイトの「ペイロード」に加えて、追加の 20 バイト IP ヘッダーが含まれる）。 1500 バイト、72 バイト、および 120 バイトの 3 つのパケットが、IPsec + GRE ピアに転送されます。
6. 受信側ルータでは、元の 1552 バイトの IPsec + GRE パケットを取得するため、2 つの IPsec フラグメント（1500 バイトと 72 バイト）が再構成されます。 120 バイトの IPsec + GRE パケットに対しては、必要な処理はありません。
7. 1500 バイトと 68 バイトの GRE パケットを取得するため、IPsec で 1552 バイトと 120 バイトの両方の IPsec + GRE パケットが復号化されます。

8. GRE では、1476 バイトと 44 バイトの IP パケット フラグメントを取得するため、1500 バイトと 68 バイトの GRE パケットがカプセル化解除されます。 これらの IP パケット フラグメントが、宛先ホストに転送されます。
9. Host 2 は、元の 1500 バイトの IP データグラムを取得するため、これらの IP フラグメントを再構成します。

シナリオ 10 はシナリオ 8 に類似していますが、トンネル パスにより低い MTU リンクが存在している点が異なります。 これは、Host 1 から Host 2 に最初のパケットを送信する場合の「ワーストケース」シナリオです。 このシナリオの最後の手順を完了すると、Host 1 は Host 2 の正しい PMTU を設定し、Host 1 と Host 2 との間の TCP 接続はすべて良好となります。Host 1 と他のホスト間の（IPsec + GRE トンネルを介して到達可能な）TCP フローに関しては、シナリオ 10 の最後の 3 つの手順だけを実行する必要があります。

このシナリオでは、**tunnel path-mtu-discovery** コマンドが GRE トンネルに設定され、Host 1 で生成される TCP/IP パケット上に DF ビットが設定されます。

シナリオ 10



1. ルータは、1500 バイトのパケットを受信します。 このパケットは GRE によって廃棄され

ます。この理由は、DF ビットが設定されているので GRE ではパケットのフラグメント化または転送を実行できず、GRE オーバーヘッド（24 バイト）が追加されると、パケット サイズが発信インターフェイスの「ip mtu」を超過するからです。
2. ルータは、ネクスト ホップの MTU が 1476（1500 - 24 = 1476）であることを通知するため、ICMP メッセージを Host 1 に送ります。
3. Host 1 は Host 2 の PMTU を 1476 に変更し、パケットを再送信する場合に、より小さいサイズで送信します。 GRE でこれがカプセル化され、1500 バイトのパケットが IPsec にわたされます。 IPsec はパケットを廃棄します。この理由は、GRE によって（設定状態の）DF ビットが内側の IP ヘッダーからコピーされており、IPsec オーバーヘッド（最大 38 バイト）を付加したパケットは大きすぎて、物理インターフェイスから転送できないためです。
4. IPsec は ICMP メッセージを GRE に送信し、ネクスト ホップの MTU が 1462 バイト（暗号化と IP オーバーヘッドに最大 38 バイトが追加されるため）であることを通知します。 GRE では、値 1438（1462 - 24）をトンネル インターフェイス上の「ip mtu」として記録します。**注:** この値の変更は内部的に格納されており、**show ip interface tunnel<#>** コマンドの出力には表示されません。 一方、**debug tunnel** コマンドを使用すると、この変更が表示されます。
5. 次に Host 1 が 1476 バイトのパケットを再送信すると、GRE はそれを廃棄します。
6. このルータは、ネクスト ホップの MTU が 1438 であることを通知する ICMP メッセージを Host 1 に送ります。
7. Host 1 は Host 2 の PMTU を低下させ、1438 バイトのパケットを再送信します。 今回は、GRE はパケットを受け入れてカプセル化し、暗号化のために IPsec にわたします。 IPsec パケットは中継ルータに転送されますが、中継ルータの発信インターフェイス MTU が 1400 なので、廃棄されます。
8. 中継ルータは、ネクスト ホップの MTU が 1400 であることを通知する ICMP メッセージを IPsec に送信します。 この値は、関連する IPsec SA の PMTU 値内に IPsec により記録されます。
9. Host 1 が 1438 バイトのパケットを再送信すると、GRE はこれをカプセル化して IPsec にわたします。 IPsec はその PMTU を 1400 に変更しているので、このパケットを廃棄します。
10. IPsec は ICMP エラーを GRE に送信し、ネクスト ホップの MTU が 1362 であることを通知し、GRE はこの値 1338 を内部に記録します。
11. Host 1 が（確認応答を受け取っていないため）元のパケットを再送信すると、GRE はこれを廃棄します。
12. ルータは、ネクスト ホップの MTU が 1338（1362 - 24 バイト）であることを通知する ICMP メッセージを Host 1 に送信します。 Host 1 は、Host 2 のための PMTU を 1338 に低下させます。
13. Host 1 は、1338 バイトのパケットを再送信し、今回は、このパケットは最終的に Host 2 まで到着できます。

その他の推奨事項

GRE と IPsec が同じルータ上に設定されている場合、トンネル インターフェイス上で **tunnel path-mtu-discovery** コマンドを設定することは、それらのインタラクションに有用です。 **tunnel path-mtu-discovery** コマンドを設定していないと、GRE IP ヘッダー内の DF ビットが常にクリアされることに注意してください。 これにより、カプセル化されたデータ IP ヘッダーで DF ビットが設定されていた場合（この場合、通常はパケットのフラグメント化が許可されません）でも、GRE IP パケットのフラグメント化が許可されます。

tunnel path-mtu-discovery コマンドが GRE トンネル インターフェイスで設定されていると、次の状態になります。

1. GRE では、データ IP ヘッダーから GRE IP ヘッダーに、DF ビットをコピーします。
2. GRE IP ヘッダー内で DF ビットが設定されていると、IPsec 暗号化後のパケットが物理発信インターフェイスの IP MTU に対して「大きすぎる」場合、IPsec はそのパケットを廃棄し、GRE トンネルに IP MTU サイズを縮小するように通知します。
3. IPsec はそれ自体のパケットに対して PMTUD を実行しますが、IPsec PMTU が変更（縮小）されても、これは即座には、IPsec から GRE に通知されません。ところが、別の「大きすぎる」パケットが到着すると、ステップ 2 の処理が発生します。
4. この場合、GRE の IP MTU はさらに小さいので、DF ビットが設定された大きすぎるデータ IP パケットをすべて廃棄し、送信側ホストに ICMP メッセージを送信します。

ip mtu コマンドを使用した静的な設定とは異なり、**tunnel path-mtu-discovery** コマンドは、GRE インターフェイスが IP MTU を動的に設定するのに有効です。 実際には、両方のコマンドの使用が推奨されます。 **ip mtu** コマンドは、ローカルの物理発信インターフェイスの IP MTU に関連する、GRE と IPsec のオーバーヘッドのためのスペースを確保するのに使用されます。 **tunnel path-mtu-discovery** コマンドでは、IPsec ピア間のパスにもっと低い IP MTU のリンクが存在する場合に、GRE トンネルの IP MTU をさらに低下させることができます。

GRE + IPsec トンネルが設定されたネットワーク内で、PMTUD に問題が発生する場合に実行可能な対応を、次に示します。

次のリストでは、最も推奨されるソリューションから掲載しています。

- PMTUD が機能しない問題を解決します。この問題は通常、ICMP をブロックしているルータまたはファイアウォールが原因です。
- そのトンネル インターフェイスに **ip tcp adjust-mss** コマンドを使用して、ルータが TCP SYN パケットの TCP MSS 値を低下させるようにします。 これは 2 つのエンドホスト（TCP の送信側および受信側）で、PMTUD が必要とされないくらい小さいパケットを使用する場合に有効です。
- ルータの入力側インターフェイスでポリシー ルーティングを使用し、さらに、ルート マップを設定して、データ IP ヘッダー内の DF ビットが GRE トンネル インターフェイスに到着する前にクリアされるようにします。 これにより、データ IP パケットを、GRE カプセル化の前にフラグメント化できるようになります。
- 発信インターフェイスの MTU と等しくなるように、GRE トンネル インターフェイスの「ip mtu」を増加させます。 これにより、フラグメント化を先に実行しないでも、データ IP パケットの GRE でのカプセル化ができるようになります。 次に、GRE パケットに対して IPsec 暗号化が実行され、物理発信インターフェイスから送信するためにフラグメント化されます。 この場合、GRE トンネル インターフェイスで **tunnel path-mtu-discovery** コマンドの設定は行いません。 これによりスループットが極端に下がる場合があります。これは、IPsec ピアでの IP パケットの再構成がプロセス交換モードで実行されることが原因です。

関連情報

- IP ルーティングに関するサポート ページ
- IPSec（IP セキュリティ プロトコル）に関するサポート ページ
- IPSec オーバーヘッド カルキュレータ（IPSec カプセル化プロトコルでパケット サイズを計算する）

- RFC 1191 Path MTU Discovery
- RFC 1063 IP MTU Discovery オプション
- RFC 791 インターネット プロトコル
- RFC 793 Transmission Control Protocol
- RFC 879 The TCP Maximum Segment Size and Related Topics
- RFC 1701 Generic Routing Encapsulation（GRE）
- RFC 1241 A Scheme for an Internet Encapsulation Protocol
- RFC 2003 IP Encapsulation within IP
- テクニカルサポート - Cisco Systems