GRE および IPsec での IP フラグメンテーション、MTU、MSS、PMTUD の問題の解決

内容

はじめに

このドキュメントでは、IPv4 フラグメンテーションと Path Maximum Transmission Unit Discovery（PMTUD）の仕組みについて説明します。

背景説明

また、IPv4 トンネルのさまざまな組み合わせとともに使用される PMTUD の動作に関連するシナリオについても説明します。

IPv4 フラグメンテーションおよび再構成

IPv4 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 Maximum Transmission Unit (MTU; 最大伝送ユニット) 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、伝送リンクによって異なります。

IPv4 では、ルータによる IPv4 データグラムのフラグメンテーションを必要に応じて許可するため、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 IPv4 データグラムにリアセンブルする役割があります。

IPv4 フラグメンテーションにより、データグラムは、後でリアセンブルされる断片に分割されます。

IPv4 のフラグメンテーションとリアセンブルには、IPv4 ヘッダーの「More Fragment」(MF) フラグおよび「Do Not Fragment」(DF) フラグとともに、IPv4 送信元、宛先、ID、合計長、およびフラグメント オフセット フィールドが使用されます。

Pv4 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。

図は IPv4 ヘッダーのレイアウトを示しています。

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

識別番号は、IPv4 データグラムの送信元によって割り当てられる 16 ビットの値です。これは、データグラムのフラグメントのリアセンブルに役立ちます。

フラグメント オフセットは 13 ビットであり、元の IPv4 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。

IPv4 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。「Do Not Fragment」(DF) ビットによって、パケットのフラグメント化が許可されるかどうかが決定さ

れます。

ビット 0 は予約済みで、常に 0 に設定されています。

ビット 1 は DF ビットです（0 =「Can 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	姓
1	0	Do not	その他（More）

IPv4 フラグメントの長さを合計すると、その値は、元の IPv4 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 IPv4 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のオフセットは 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 IPv4 ヘッダーの 20 バイトが含まれます。

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。つまり、このフラグメントのデータ部分は、元の IPv4 データグラムの 1480 バイト目から始まります。

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 IPv4 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。つまり、このフラグメントのデータ部分は、元の IPv4 データグラムの 2960 バイト目から始まります。

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 IPv4 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 IPv4 データグラムの 4440 バイト目から始まるという意味です。

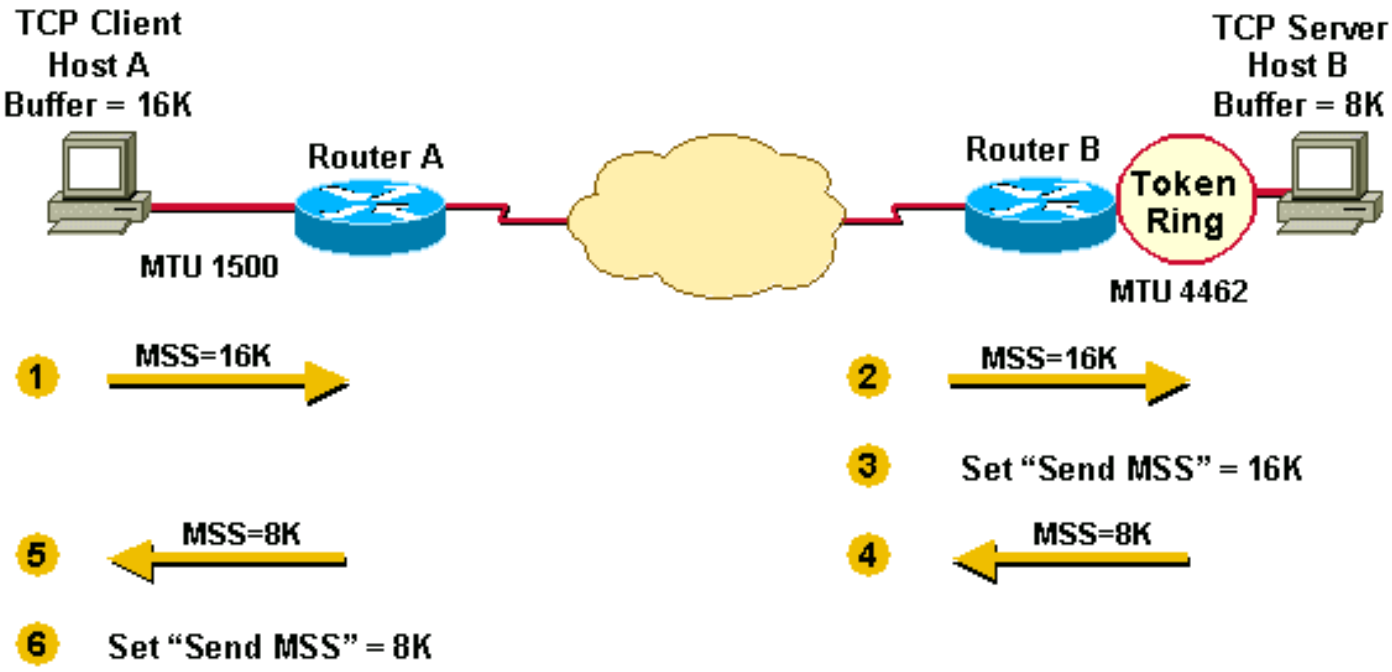
このフラグメントの長さは 700 バイトです。これには、このフラグメントのために作成された追加の IPv4 ヘッダーが含まれます。

元の IPv4 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 IPv4 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。

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 IPv4 データグラムのデータ部分である 5120 バイトになります。

図に示すように、IPv4 ヘッダーの 20 バイトが追加されると、元の IPv4 データグラムのサイズ（4440 + 680 + 20 = 5140）と等しくなります。



IPv4 フラグメンテーションに関する問題

IPv4 フラグメンテーションにより、IPv4 データグラムをフラグメント化するために CPU とメモリのオーバーヘッドがわずかに増加します。これは、送信側と、送信側と受信側の間のパスに含まれるルータに当てはまります。

フラグメントの作成にはフラグメントヘッダーの作成が含まれ、元のデータグラムがフラグメントにコピーされます。

フラグメントの作成に必要なすべての情報はすぐに利用できるので、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。

ただし、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でのリアセンブルは、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。

リアセンブルを実行するルータでは、使用可能な最大バッファ（18K）が選択されます。これは、最後のフラグメントが受信されるまで、元の IPv4 パケットのサイズが分からないからです。

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。

IPv4 データグラムの 1 つのフラグメントがドロップされると、元の IPv4 データグラム全体が存

在する必要があり、それが再度フラグメント化されます。

これは、ネットワーク ファイル システム（NFS）で発生します。NFS には 8192 の読み取りおよび書き込みブロックサイズが用意されます。

そのため、NFS IPv4/UDP データグラムは約 8500 バイトになります（NFS、UDP、および IPv4 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と 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 6 つのフラグメントのいずれかがドロップされた場合は、元のデータグラム全体を再転送する必要があります。これにより、さらに 6 つのフラグメントが作成されます。

このリンクで 6 つのパケットのうち 1 つがドロップされると、各 NFS で元の 8500 バイトの IPv4 データグラムから少なくとも 1 つの IPv4 フラグメントがドロップされるため、このリンクを介して NFS データが転送される可能性は低くなります。

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処理または操作するファイアウォールでは、IPv4 フラグメントを適切に処理できません。

IPv4 フラグメントの順序が正しくない場合は、1 番目以外のフラグメントがファイアウォールによってブロックされます。これは、そ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 IPv4 データグラムのリアセンブルが不可能であることを意味します。

含まれる情報が不十分なフラグメント（1 番目以外のフラグメント）がフィルタと適切に合致すること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ってファイアウォールを突破する攻撃が可能になります。

コンテンツスイッチエンジンなどのネットワークデバイスは、L4 から L7 の情報に基づいてパケットを転送します。パケットが複数のフラグメントにわたる場合、それらのデバイスはポリシーの適用に失敗します。

IPv4 フラグメンテーションの回避：TCP MSS の仕組み

単一の TCP/IPv4 データグラムでは、TCP の最大セグメントサイズ（MSS）により、ホストが受け入れる最大データが定義されます。

この TCP/IPv4 データグラムは、IPv4 レイヤでフラグメント化されている可能性があります。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。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 IPv4 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。

MSS は TCP の受信側で受け取るデータの最大セグメントでした。この TCP セグメントは最大 64K のサイズになり、受信側ホストに転送するために IPv4 レイヤでフラグメント化されます。

受信側ホストでは、IPv4 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。

TCP セグメントと IPv4 データグラムのサイズを制限するために MSS 値を設定および使用する方法

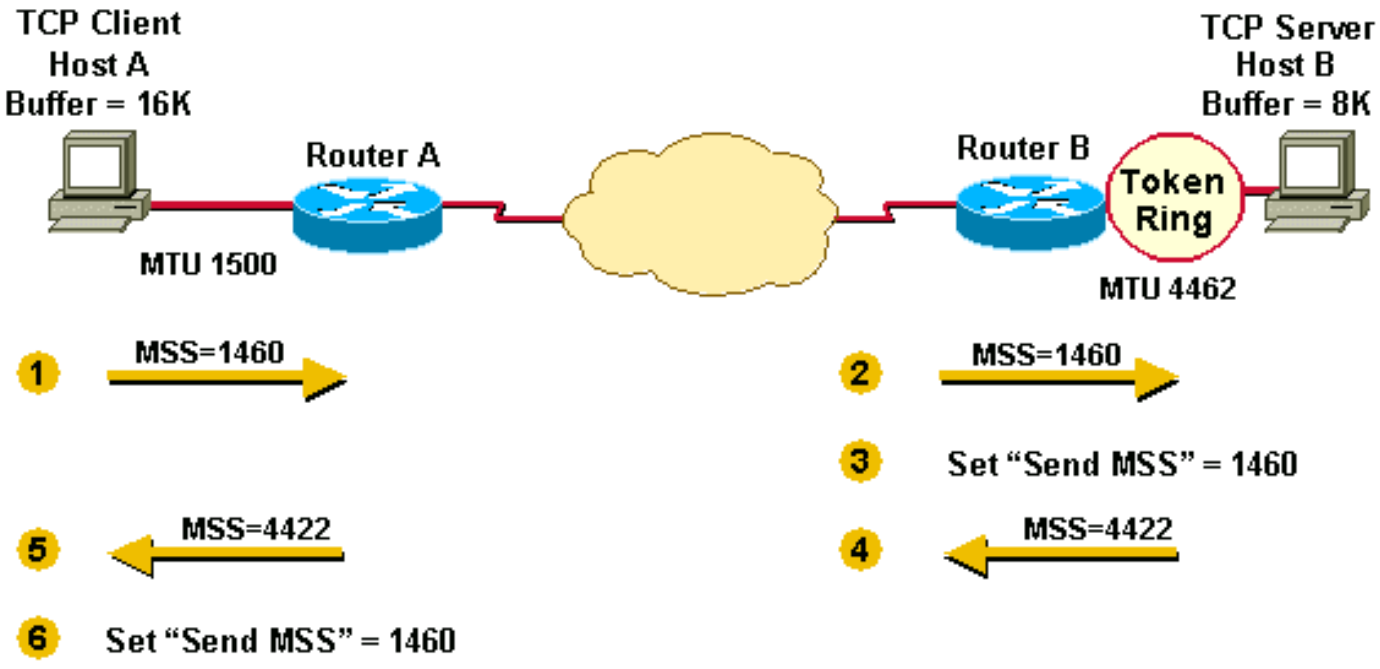
例 1 は、MSS の初期実装方法を示しています。

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。

ホスト A とホスト B では、インターフェイス MTU より大きい（ただし、送信 MSS より小さい）IPv4 データグラムをフラグメント化する必要があります。これは、TCP スタックにより、スタックにある 16K または 8K バイトのデータが IPv4 に渡されるためです。

ホスト B の場合、パケットは、トークンリング LAN に到達するためにフラグメント化され、イーサネット LAN に到達するために再度フラグメント化されます。

例 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 IPv4 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送信インターフェイスの MTU (-40) に変更されました。

MSS (TCP データサイズ) には 20 バイトの IPv4 ヘッダーと 20 バイトの TCP ヘッダーが含まれていないため、MSS の数値は MTU の数値よりも 40 バイト小さくなります。

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 IPv4 オプションに応じて、IPv4 ヘッダーおよび TCP ヘッダーのための適切な値を減算する必要があります。

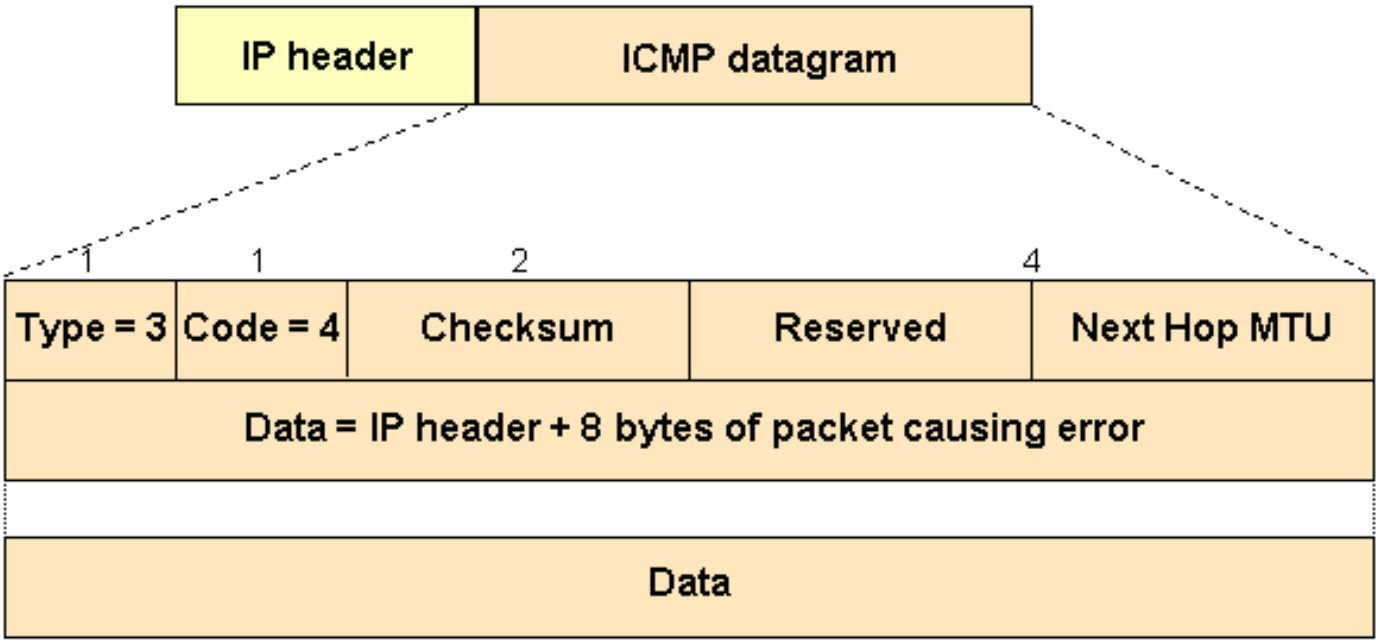
現在、MSS は、各ホストがまず送信インターフェイス MTU をそれぞれのバッファと比較し、送信する MSS として最小の値を選択するという方法で動作します。

その後、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 と比較し、2 つの値のうち小さい方を再度選択します。

例 2 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の手順を示しています。

ホストが相互に MSS 値を送信する前に、各ホストによって送信インターフェイス MTU が考慮されます。これが、フラグメンテーションの回避に役立ちます。

例 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. ホスト B は、ホスト A の送信 MSS（1460）を受信し、それを送信インターフェイス MTU - 40 の値（4422）と比較します。
3. Host B は、Host A に送信する IPv4 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. ホスト A は、ホスト B の送信 MSS（4422）を受信し、それを送信インターフェイス MTU - 40 の値（1460）と比較します。
6. Host A は、Host B に送信する IPv4 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 接続の両側で同じになります。

例 2 では、両方の送信インターフェイス MTU がホストによって考慮されるため、TCP 接続のエンドポイントでフラグメンテーションは発生しません。

ただし、いずれかのホストの送信インターフェイス MTU よりも小さい 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依然としてフラグメント化されます。

PMTUD とは

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処理しますが、これら 2 つのエンドポイントの間により小さい MTU リンクがある場合は処理しません。

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、パケットの送信元から宛先までのパスに沿った最小の 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 がホストで有効になっている場合、ホストからのすべての TCP および 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、この MTU 値を含むルーティングテーブル内にホスト（/32）エントリを作成するため、宛先の MTU 値を記録します。

ルータが、パケットサイズより小さい MTU を持つリンクへの IPv4 データグラム（DF ビットが設定された）の転送を試みると、そのルータはパケットをドロップし、「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 IPv4 データグラムの送信元に返します。

送信側ステーションは、ICMP メッセージを受信すると、送信 MSS を小さく、TCP がセグメントを再送信するときに、より小さいセグメントサイズを使用します。

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ICMP「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。

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

RFC 1191によると、「Fragmentation Needed 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、PMTUD 中に MTU を下げる推奨値を示した表も含まれています。

この例に示すように、送信 MSS の適切な値でより迅速に到達するために、これがホストによって使用されます。

送信側と受信側の間のパスは動的に変化する可能性があるため、PMTUD は、すべてのパケットに対して継続的に実行されます。

送信側が「Cannot 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。

1. パケットがフラグメント化されずに受信側まで到達する。

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 到達不能メッセージの数が 1 秒あたり 2 件に制限されます。したがって、このコンテキストでは、ルータが1秒あたり2つ以上のICMPメッセージ (タイプ= 3、コード= 4) (異なるホストの可能性あり) で応答する必要があると予想されるネットワークシナリオの場合は、 **no ip icmp rate-limit unreachable [df] interface**コマンドでICMPメッセージのスロットリングを無効にします。

---

2. 送信側では、「Cannot Fragment」ICMP メッセージを、受信側へのパス上のホップから受け取ります。

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 によって片方のエンドステーションでの送信 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 が保持されます (PMTUD をトリガーするだけの大きさを持つ IPv4 データグラムが送信されていないため) 。

例 3 に示す HTTP 接続は、その一例です。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、サーバーからの大きなパケット (576 バイトを超える) だけが PMTUD をトリガーします。

クライアントからのパケットは小さく (576 バイト未満) 、MTU が 576 のリンクを通過するためにフラグメンテーションを必要としないため、PMTUD をトリガーしません。

例 3

例 4 に、1 つのパスの最小 MTU が他のパスよりも小さい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。

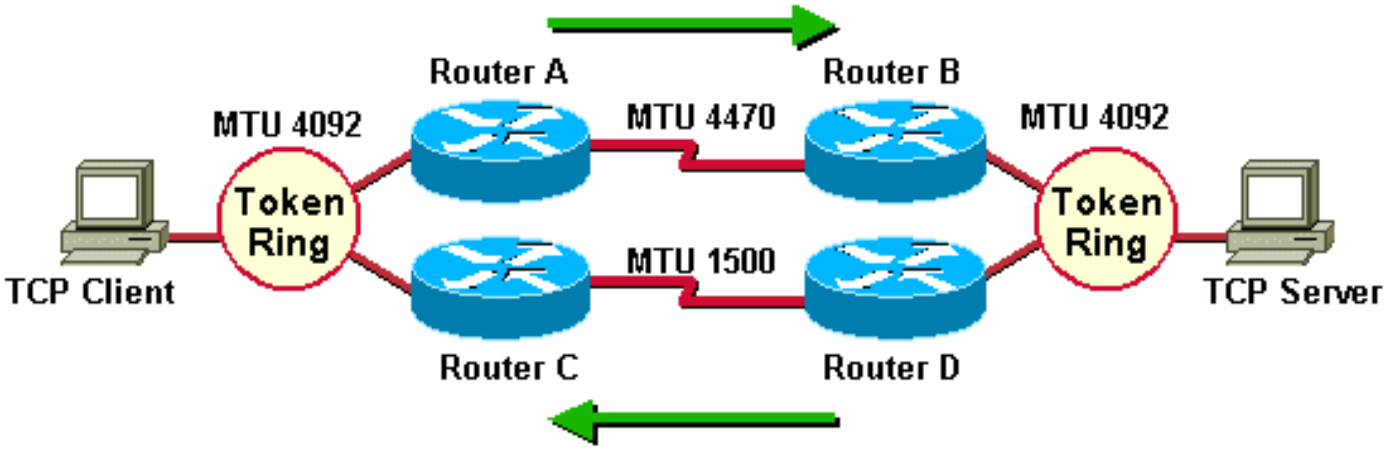
この例では、TCP フローの一方向でのみ、PMTUD によって送信 MSS の低下がトリガーされます。

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 サーバーによってクライアントにパケットが送信されると、PMTUD がサーバーによる送信 MSS の低下をトリガーします。これは、ルータ D で、ルータ C に 4092 バイトのパケットを送信する前に、パケットをフラグメント化する必要があるためです。

逆に、クライアントでは、「Fragmentation Needed and DF set」を示すコードを持つ「Destination Unreachable」 ICMP メッセージを受け取ることはありません。これは、ルータ B 経由でサーバーに送信する場合、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。

例 4



注：ip tcp path-mtu-discoveryコマンドは、ルータ（BGPやTelnetなど）によって開始されるTCP接続のTCP MTUパスディスカバリをイネーブルにするために使用されます。

PMTUD の問題

以下は、PMTUD が失敗する可能性のある状況です。

- ルータが、パケットをドロップし、ICMP メッセージを送信しない。（一般的ではない）
- ルータが ICMP メッセージを生成して送信するが、このルータと送信者の間のルータまたはファイアウォールによって ICMP メッセージがブロックされる。（一般的）
- ルータが ICMP メッセージを生成して送信するが、送信者がメッセージを無視する。（一般的ではない）

上記の 3 つのうち最初と最後は、通常、エラーの結果ですが、2 番目は、一般的な問題を示しています。

ICMP パケットフィルタの実装者は、特定の ICMP メッセージタイプだけでなく、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 メッセージタイプをブロックできます。

PMTUD の成否は、TCP/IPv4 パケットの送信側に到達する 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 IPv4 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るという問題を緩和するために使用できる他の手法があります。

- ルータ上の DF ビットをクリアし、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**>を使用して、TCP MSSのオプション値MSSを操作します。

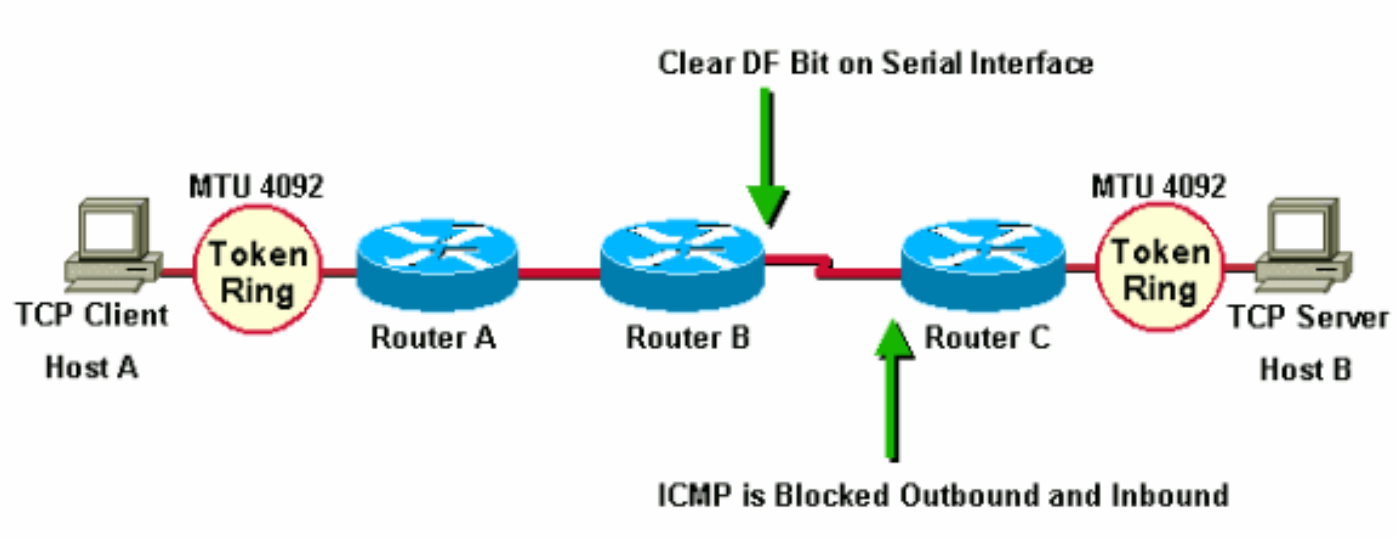
次の例では、ルータ A とルータ B が同じ管理ドメイン内にあります。ルータ C はアクセスできず、ICMP をブロックするので、PMTUD が失敗します。

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
```



別のオプションは、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(1460)が **ip tcp adjust-mss**コマンド内の値(1460)よりも小さくなります。

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。

IPv4 パケットのサイズは、TCP ヘッダー（20 バイト）と IPv4 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 コマンドを使用すると、TCP SYNパケットのMSSを調整できます。次の構文では、TCP セグメントの MSS 値が 1460 に減らされます。

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
```

IPv4 トンネルがより広く普及するようになったのに従い、IPv4 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。

トンネルのカプセル化によってパケットサイズに「オーバーヘッド」が追加されるため、トンネルでより多くのフラグメンテーションが発生します。

たとえば、Generic Router Encapsulation（GRE）の付加により、パケットに 24 バイトが追加されます。この増加によって、パケットが送信側 MTU より大きくなり、フラグメント化が必要になります。

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、IPv4sec、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、送信インターフェイスの有効な MTU を減少させます。

Tunnel（トンネル）

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 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。トンネルインターフェイスには、次の 3 つの主要コンポーネントがあります。

- パッセンジャ プロトコル（AppleTalk、Banyan VINES、CLNS、DECnet、IPv4、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。詳細は、RFC 2784 および RFC 1701 を参照してください。

  - IPv4 トンネル内の IPv4：詳細は、RFC 2003 を参照してください。

- 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

このセクションで示すパケットは、GRE がカプセル化プロトコルであり、IPv4 がトランスポート プロトコルであるという IPv4 トンネリングの概念を示しています。

また、パッセンジャ プロトコルも IPv4 です。この場合、IPv4 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

IPv4	TCP	Telnet

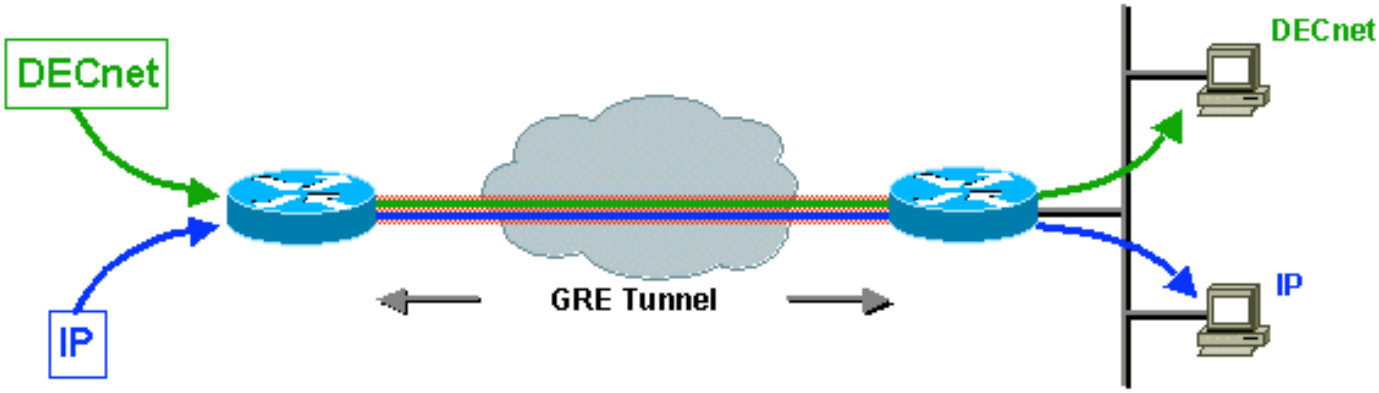
トンネル パケット

IPv4	GRE	IPv4	TCP	Telnet

- IPv4 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。
- IPv4 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 IPv4 および DECnet のカプセル化を示しています。

これは、図に示すように、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。



IPv4 バックボーンによって隔てられた 2 つの非隣接の非 IPv4 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します。

この非隣接ネットワークで DECnet が実行されている場合、管理者は、バックボーンに DECnet を設定することにより、それらを接続する（または接続しない）ことを選択できます。

管理者は、IPv4 ネットワークのパフォーマンスを低下させる可能性があるため、バックボーン帯域幅を消費する DECnet ルーティングを許可することを望みません。

実行可能な代案は、IPv4 バックボーンを介して DECnet をトンネル化することです。トンネルソリューションによって IPv4 内で DECnet パケットがカプセル化され、バックボーンを介してトンネルエンドポイントに送信されます。トンネルエンドポイントでは、カプセル化が解除され、DECnet パケットが DECnet を介して宛先にルーティングされます。

別のプロトコル内でトラフィックをカプセル化する利点は、次のとおりです。

- 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）、バックボーンはこれらのアドレスのルーティングをサポートしていません。
- 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。

以降、IPv4 がパッセンジャプロトコルとして使用され、IPv4 がトランスポートプロトコルとして使用されます。

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、プロセス交換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たトンネリングの場合は、トンネリングされるパッセンジャプロトコ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）に問題が発生します。

- 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します。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消費します。複数のポイントツーポイント トンネルを介する場合、各トンネルインターフェイスに帯域幅が割り当てられ、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ます。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は、実リンクを介したトンネルが優先されます。これは、そのトンネルが、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、その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されます。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してください。

- 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されます。再帰ルーティングとは、トンネル宛先へのベスト 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、次のエ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 参加者に関与するルータ

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 IPv4 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 IPv4 パケットに関して、ホストのような動作をします。

ルータが 1 番目の役割 (ホスト IPv4 パケットを転送するルータ) で動作する場合、この役割は、ルータがトンネルパケット内にホスト IPv4 パケットをカプセル化する前に実行されます。

ルータは、ホストパケットの転送側となる場合、次のアクションを実行します。

- 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。

このセクションでは、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** のインタラクションを示す **2** つの例について詳しく説明します。

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 を処理するために、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、適切な処理を実行する必要があります。

この例では、トンネルの GRE カプセル化を使用しています。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 IPv4 MTU は 1476（1500 - 24）です。

例 1

1. トンネル送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（DF = 0）された 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。

IPv4	1480 バイト TCP + データ

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 IPv4 MTU に対して大きくなりすぎるため、転送ルータによってデータグラムが 1476 バイト（20 バイトの IPv4 ヘッダー + 1456 バイトの IPv4 ペイロード）と 44 バイト（20 バイトの IPv4 ヘ

ッダー + 24 バイトの IPv4 ペイロード) の 2 つのフラグメントに分割されます。

GRE カプセル化が追加されても、パケットは送信物理インターフェイスの MTU を超えなくなります。

IP_0	1456 バイト TCP + データ
IP_1	24 バイト データ

3. 転送ルータでは、GRE カプセル化による追加があります。元の IPv4 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 IPv4 ヘッダーが追加分になります。

これにより、これら 2 つの IPv4 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 IPv4 データグラムとして認識されます。

IPv4	GRE	IP_0	1456 バイト TCP + データ
IPv4	GRE	IP_1	24 バイト データ

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 IPv4 フラグメントが残されます。

これらの IPv4 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。

IP_0	1456 バイト TCP + データ
IP_1	24 バイト データ

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。

IPv4	1480 バイト TCP + データ

例 2 に、ネットワークトポロジの観点での転送ルータの役割を示します。

この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. トンネル送信元の転送ルータは、送信側ホストから、DF = 1 の 1500 バイトのデータグラムを受信します。

IPv4	1480 バイト TCP + データ

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 IPv4 MTU (1476) より大きいので、ルータはデータグラムをドロップし、「ICMP Fragmentation Needed but DF Bit Set」メッセージをデータグラムの送信元に送信します。

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。

IPv4	ICMP MTU 1476

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 IPv4 データグラムを使用します。

IPv4	1456 バイト TCP + データ

4.

この IPv4 データグラムの長さ（1476 バイト）は、今回は GRE トンネル IPv4 MTU の値に等しいため、ルータはこの IPv4 データグラムに GRE カプセル化を行います。

IPv4	GRE	IPv4	1456 バイト TCP + データ

5.（トンネル宛先側の）受信側ルータは、IPv4 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。

IPv4	1456 バイト TCP + データ

これは、PMTUD およびトンネル IPv4 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です。

この役割は、ルータがトンネルパケット内に元の IPv4 パケットをカプセル化した後、実行されます。

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-IPv4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 IPv4 MTU に収まるほど小さい IPv4 データグラムをホストが送信してい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-IPv4パケットへのPMTUDは実行されません。

例 3

1. トンネル送信元の転送ルータは、送信ホストから 1476 バイトのデータグラムを受信します。

IPv4	1456 バイト TCP + データ

2. このルータでは GRE 内で 1476 バイトの IPv4 データグラムがカプセル化されて、1500 バイトの GRE IPv4 データグラムが作成されます。

GRE IPv4 ヘッダー内の DF ビットはクリア（DF = 0）されます。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。

IPv4	GRE	IPv4	1456 バイト TCP + データ

3. トンネルの送信元と宛先の間に、リンク MTU が 1400 のルータが存在すると仮定します。

DF ビットがクリア（DF = 0）されているので、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 IPv4 がフラグメント化されるので、GRE、内側の IPv4、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。

IP_0	GRE	IP	1352 バイト TCP + データ
IP_1	104 バイト データ		

4. このトンネルの宛先ルータで、GRE トンネルパケットをリアセンブルする必要があります。

IP	GRE	IP	1456 バイト TCP + データ

5. GRE トンネルパケットがリアセンブルされると、ルータは GRE IPv4 ヘッダーを削除し、元の IPv4 データグラムをその宛先に送信します。

IPv4	1456 バイト TCP + データ

例 4 では、PMTUD およびトンネル IPv4 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。

今回は、元のIPv4ヘッダー内でDFビットが設定され(DF = 1)、内側のIPv4ヘッダーから外側の(GRE + IPv4)ヘッダーにDFビットがコピーされるように **tunnel path-mtu-discovery** コマンドが設定されています。

例 4

1. トンネル送信元の転送ルータは、送信側ホストから、DF = 1 の 1476 バイトのデータグラムを受信します。

IPv4	1456 バイト TCP + データ

2. このルータでは GRE 内で 1476 バイトの IPv4 データグラムがカプセル化されて、1500 バイトの GRE IPv4 データグラムが作成されます。

元の IPv4 データグラムの DF ビットが設定されているため、この GRE IPv4 ヘッダーでは DF ビットが設定 (DF = 1) されます。

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。

IPv4	GRE	IPv4	1456 バイト TCP

3. 再び、トンネルの送信元と宛先の間に、リンク MTU が 1400 のルータが存在すると仮定します。

DF ビットが設定 (DF = 1) されているので、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ドロップし、ICMP エラーメッセージをトンネルの送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送信元 IPv4 アドレスであるからです。

IPv4	ICMP MTU 1400

4. トンネル送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 IPv4 MTU を 1376 (1400 - 24) に減少させます。

送信側ホストが次にデータを 1476 バイトの IPv4 パケットで再送信すると、このパケットは大きくなりすぎる場合があり、その後、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。

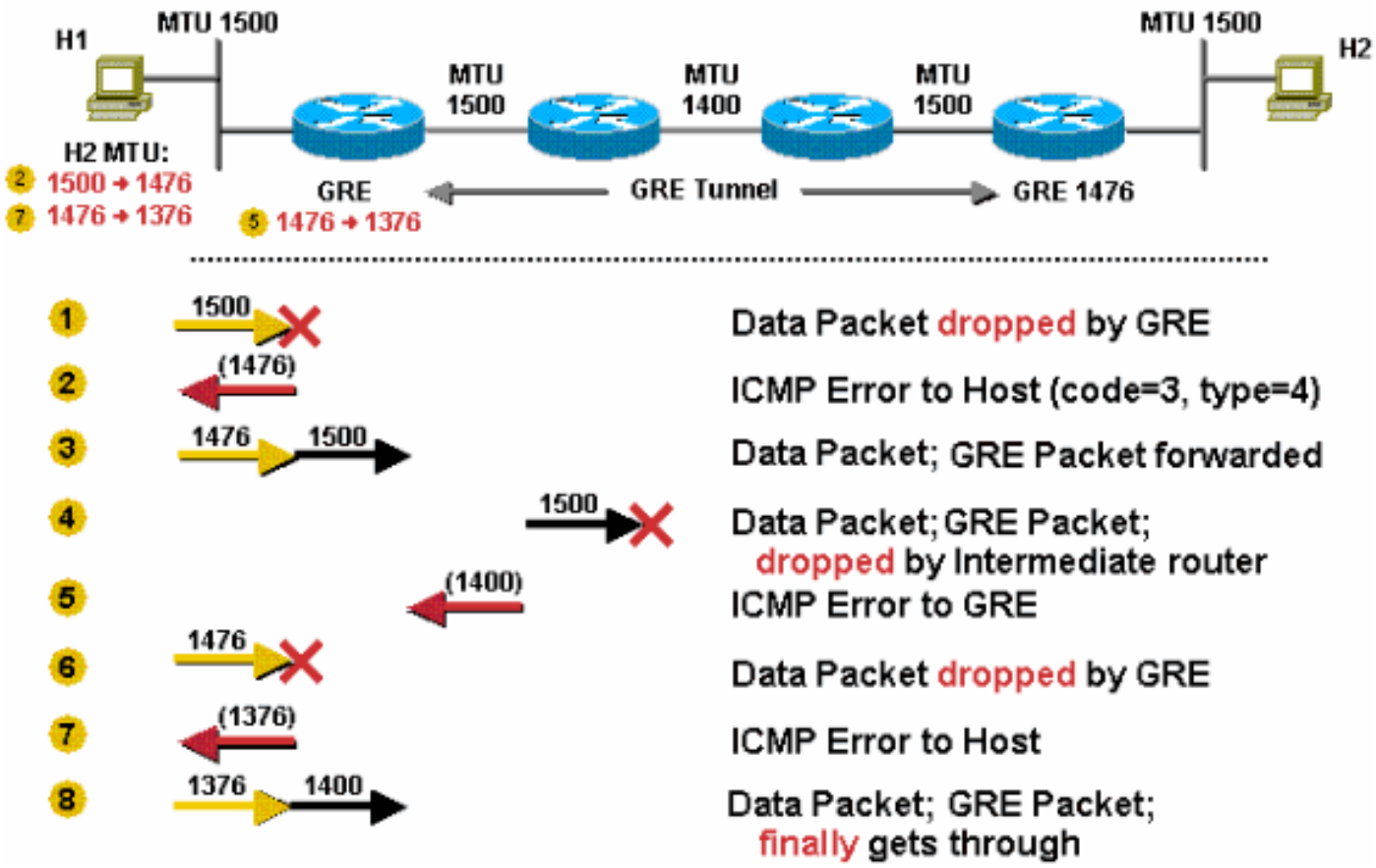
送信側ホストがデータを再送信する際には、1376 バイトの IPv4 パケットで送信し、このパケットは GRE トンネルを介して受信側ホストに到着します。

例 5

この例では、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、その後、データパケットの

PMTUD を実行します。IPv4 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ません。

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 IPv4 MTU は、デフォルトでは物理インターフェイスの IPv4 MTU より 24 バイト少ないので、図のように、GRE インターフェイスの IPv4 MTU は 1476 となります。



- 送信者は 1500 バイトのパケット（20 バイト IPv4 ヘッダー + 1480 バイトの TCP ペイロード）を送信します。
- 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 IPv4 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。
- 24 バイトの GRE ヘッダーが各 IPv4 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。
- 2 つの IPv4 フラグメントを含む GRE + IPv4 パケットが、GRE トンネル ピア ルータに転送されます。
- 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。
- 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。
- 宛先ホストは、この IPv4 フラグメントを元の データグラムに再構成します。

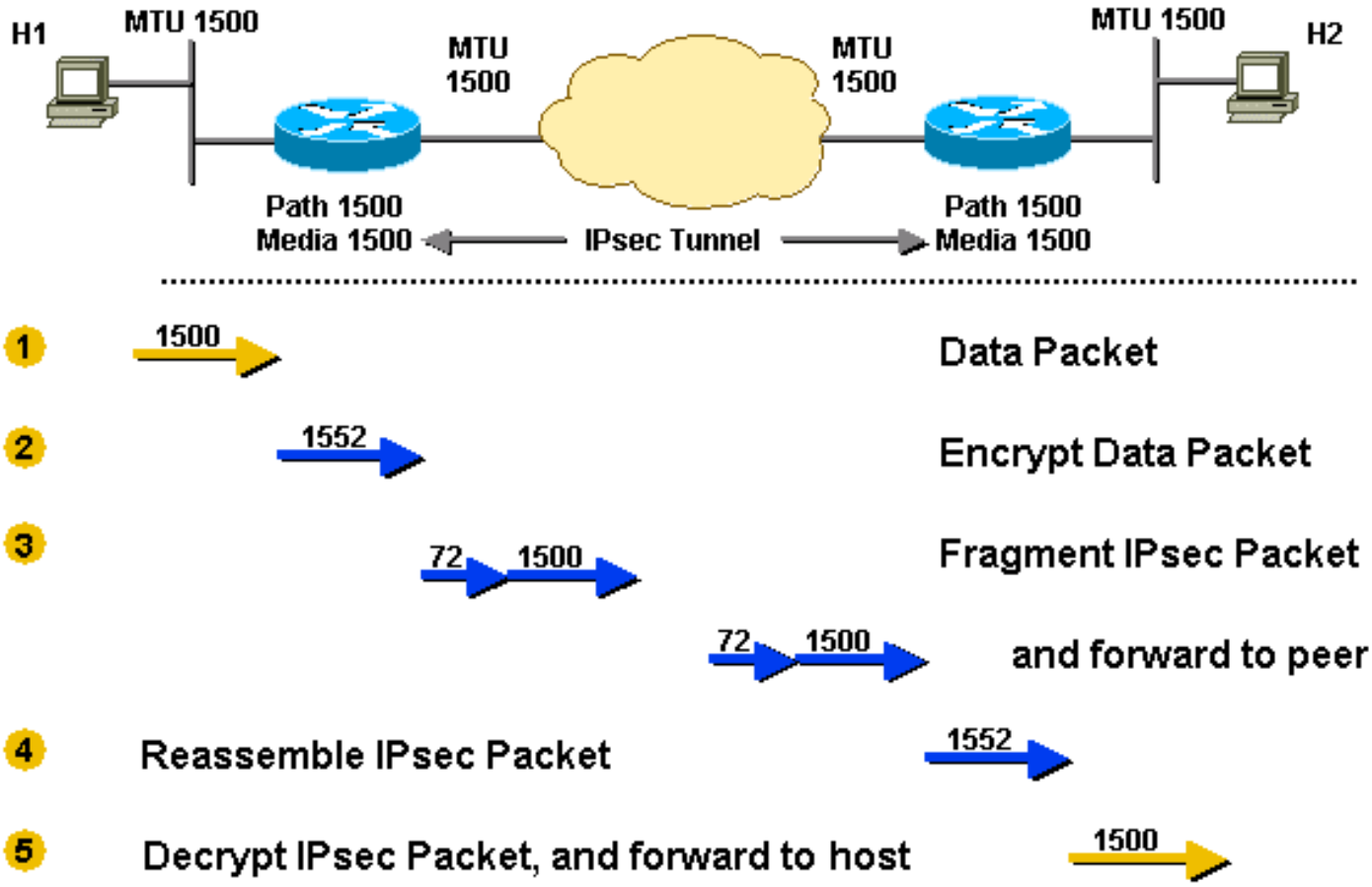
例 6

この例は例 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。ルータは、GRE + IPv4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IPv4ヘッダーからGRE IPv4ヘッ

ダーにコピーされます。

ルータでは、GRE + IPv4 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 IPv4 MTU を低下させます。

GRE トンネルの IPv4 MTU は、デフォルトでは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、ここでの GRE IPv4 MTU は 1476 となります。図に示すように、GRE トンネルパス内に MTU が 1400 のリンクが存在します。



- ルータは、1500 バイトのパケット（20 バイトの IPv4 ヘッダー + 1480 バイトの TCP ペイロード）を受信し、このパケットをドロップします。ルータがこのパケットをドロップする理由は、これが GRE トンネル インターフェイス上の IPv4 MTU（1476）よりも大きいためです。

- 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。

- 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。

- 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。

- 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 IPv4 MTU 値を設定します。この変更は、 **debug tunnel** コマンドを使用する場合にだけ表示され、 **show ip interface tunnel**<#> コマンドの出力では表示されません。

- 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

IPv4 MTU（1376）より大きいので、GRE ルータはこれをドロップします。

- 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。

- 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/IPv4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、リアセンブルする必要があります。

純粋な IPsec トンネルモード

IPv4 セキュリティ（IPv4sec）プロトコルは、IPv4 ネットワークを介して転送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、標準ベースのメソッドです。

IPv4sec では、IPv4 ネットワーク層での暗号化が提供されます。IPv4sec では、少なくとも 1 つの IPv4 ヘッダー（トンネル モード）が追加されるので、IPv4 パケットが長くなります。

追加されるヘッダーの長さは、IPsec の設定モードによって異なります。ただし、パケットにつき 58 バイト（Encapsulating Security Payload（ESP）および ESP authentication（ESPauth））を超えることはありません。

IPv4sec 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 IPv4 パケットはすべて保護され（暗号化、認証、またはその両方）、IPv4sec 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 IPv4 ヘッダーがパケットの先頭に付加されて、送信元および宛先に IPv4sec 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 IPv4 トラフィックで使用でき、IPv4sec がホストからのトラフィックを IPv4sec 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 モードはバーチャル プライベート ネットワーク（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 IPv4sec ピアが経由されます。VPN では、IPv4sec「トンネル」は IPv4sec 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 IPv4 トラフィックを保護します。

- トランスポートモード(トランスフォーム定義でサブコマンド **mode transport**を使用して設定)では、元のIPv4パケットのペイロードだけが保護されます（暗号化、認証、またはその両方）。ペイロードは、IPv4sec のヘッダーとトレーラでカプセル化されます。IPv4 プロトコル フィールドの ESP（50）への変更を除き、元の IPv4 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、パケットが復号される際の復元のために IPv4sec トレーラに保存されます。トランスポート モードは、IPv4 トラフィックが IPv4sec ピア自体の間で保護される場合にだけ使用され、パケット上の送信元および宛先の IPv4 アドレスは、IPv4sec ピア アドレスと同じになります。IPv4sec トランスポート モードが使用されるのは、通常、最初の IPv4 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 IPv4sec 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。

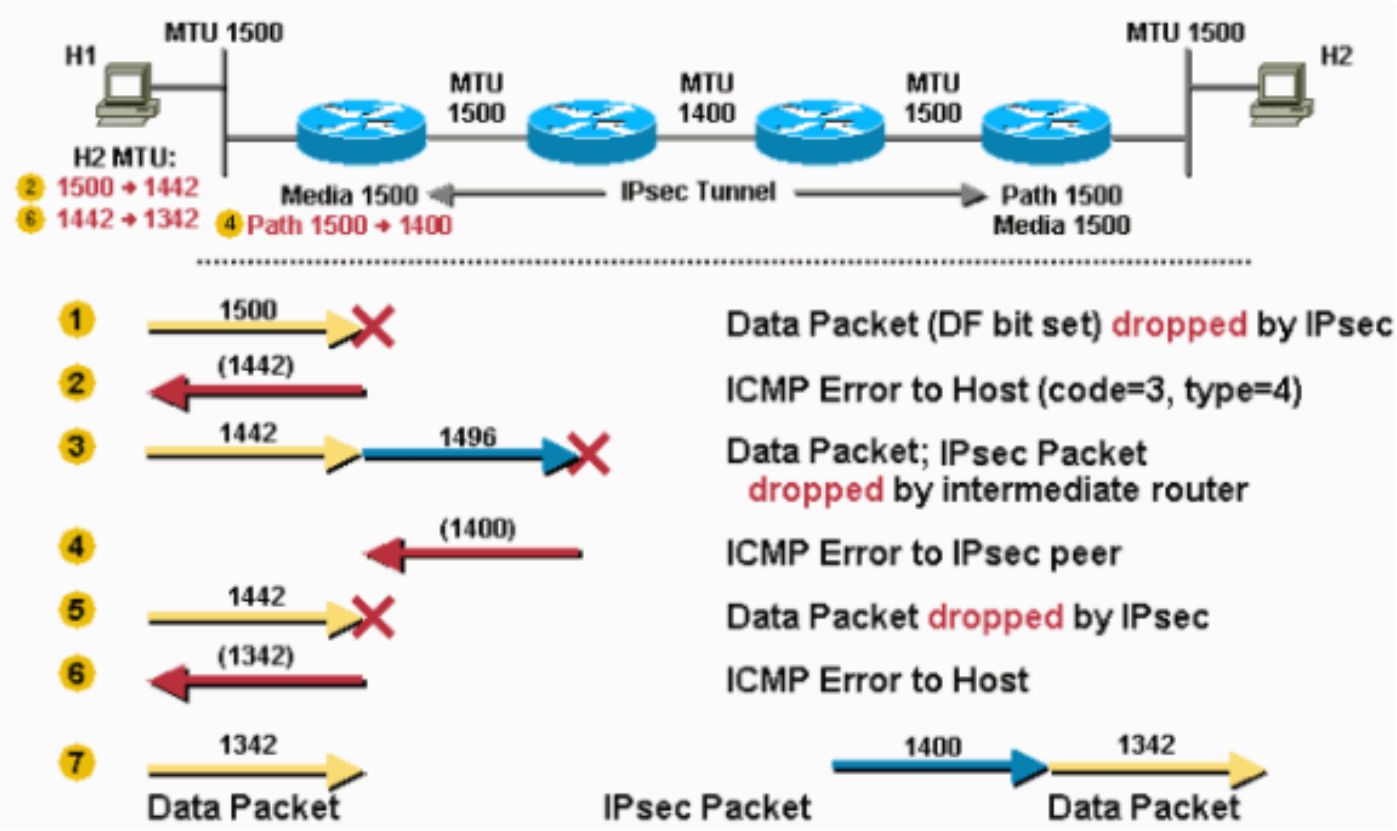
IPv4sec では常に、データ パケットおよび IPsec 自体のパケットのために PMTUD が実行されます。IPv4sec IPv4 パケットの PMTUD 処理を変更するために、IPv4sec 設定コマンドがあります。IPv4sec では DF ビットに関して、クリア、設定、またはデー

タ パケットの IPv4 ヘッダーから IPv4sec の IPv4 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：IPv4sec でハードウェアでの暗号化を行う場合は、カプセル化の後のフラグメンテーションを回避してください。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、IPv4sec 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 IPv4sec パケットが再構成のためにプロセス交換された後、復号のためにハードウェア暗号化エンジンに渡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。

例 7

このシナリオでは、実行中の IPv4sec 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。



- 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 IPv4 ヘッダー + 1480 バイトの TCP ペイロード) を受信します。

- 1500 バイトのパケットが IPv4sec によって暗号化され、52 バイトのオーバーヘッド (IPv4sec ヘッダー、トレーラ、および追加の IPv4 ヘッダー) が追加されます。これにより、IPv4sec 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。

- この IPv4sec 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 IPv4 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 IPv4 フラグメントになります。

- IPv4sec 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 IPv4 ヘッダーを取り除き、さらに、これらの IPv4 フラグメントを結合して元の IPv4sec パケットに戻します。次に、IPv4sec によってこのパケットが復号されます。

- 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。

例 8

この例は例 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、IPv4sec 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。

この例では、「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 IPv4sec ピアルータの動作を説明します。

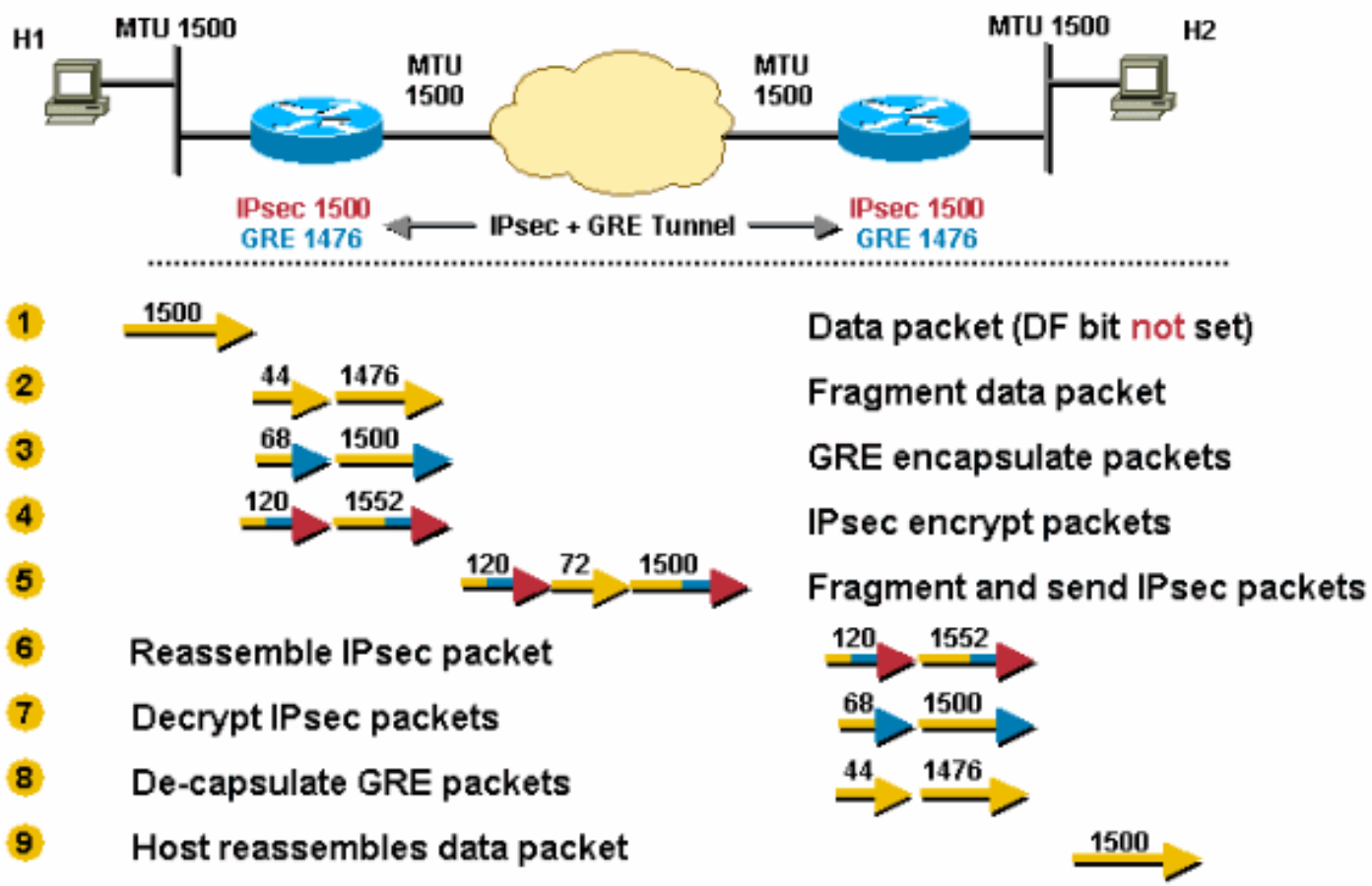
フラグメンテーションの必要性の結果として、IPv4sec PMTU が低い値に変更されます。

IPv4sec でパケットが暗号化されると、DF ビットが内側の IPv4 ヘッダーから外側の IPv4 ヘッダーにコピーされます。

メディア MTU および PMTU の値は、IPv4sec 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、アウトバウンド 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、IPv4sec ピア間のパスで発生する最小 MTU に基づいています。

図に示すように、IPv4sec では、フラグメント化が試みられる前にパケットがカプセル化/暗号化されます。



- ルータは、1500 バイトのパケットを受信してドロップします。これは、IPv4sec 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。

- 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 バイトは、IPv4sec ESP および ESPauth を使用する場合の最大 IPv4sec オーバーヘッドです。実際の IPv4sec オーバーヘッドは、この値よりも 7 バイト小さい値となる可能性があります。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。

- ホスト 1 は、ホスト 2 に対する PMTU を 1442 に減少させるので、データをホスト 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、IPv4sec 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 IPv4sec 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。

- パケットをドロップするこの中間ルータは、ICMP メッセージを IPv4sec パケットの送信者（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、IPv4sec SA PMTU 内に記録されます。

- 次にホスト 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、IPv4sec はパケットをドロップします。ルータはパケットをドロップします。これは、IPv4sec 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）より大きくなるためです。

- 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。ホスト 1 は、この情報を再度記録します。

- ホスト 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、IPv4sec トンネルを介してホスト 2 に到達します。

GRE と IPv4sec の使用

GRE トンネルの暗号化に IPv4sec 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。

IPv4sec では IPv4 マルチキャスト パケットがサポートされていないため、IPv4sec と GRE が次の方法で組み合されることになります。これは、IPv4sec 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 IPv4 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 IPv4sec 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 IPv4sec がトランスポートモードで展開されます。これは、IPv4sec ピアと GRE トンネルのエンドポイント（ルータ）が同じものであり、トランスポートモードでは IPv4sec オーバーヘッドの 20 バイトが節約されるためです。

注目すべき状況の 1 つに、IPv4 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、IPv4sec は、2 つの独立した GRE + IPv4 パケットを認識します。多くの場合、デフォルト設定では、これらのパケットの 1 つが十分に大きいため、暗号化さらた後でフラグメント化される必要があります。

IPv4sec ピアは、復号化の前にこのパケットをリアセンブル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、IPv4sec 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。

リアセンブル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 IPv4sec 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、送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 + IPv4sec（トランスポート モード）	1440 バイト	1400 バイト
GRE + IPv4sec（トンネル モード）	1420 バイト	1400 バイト

注：一般的な GRE + IPv4sec 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。

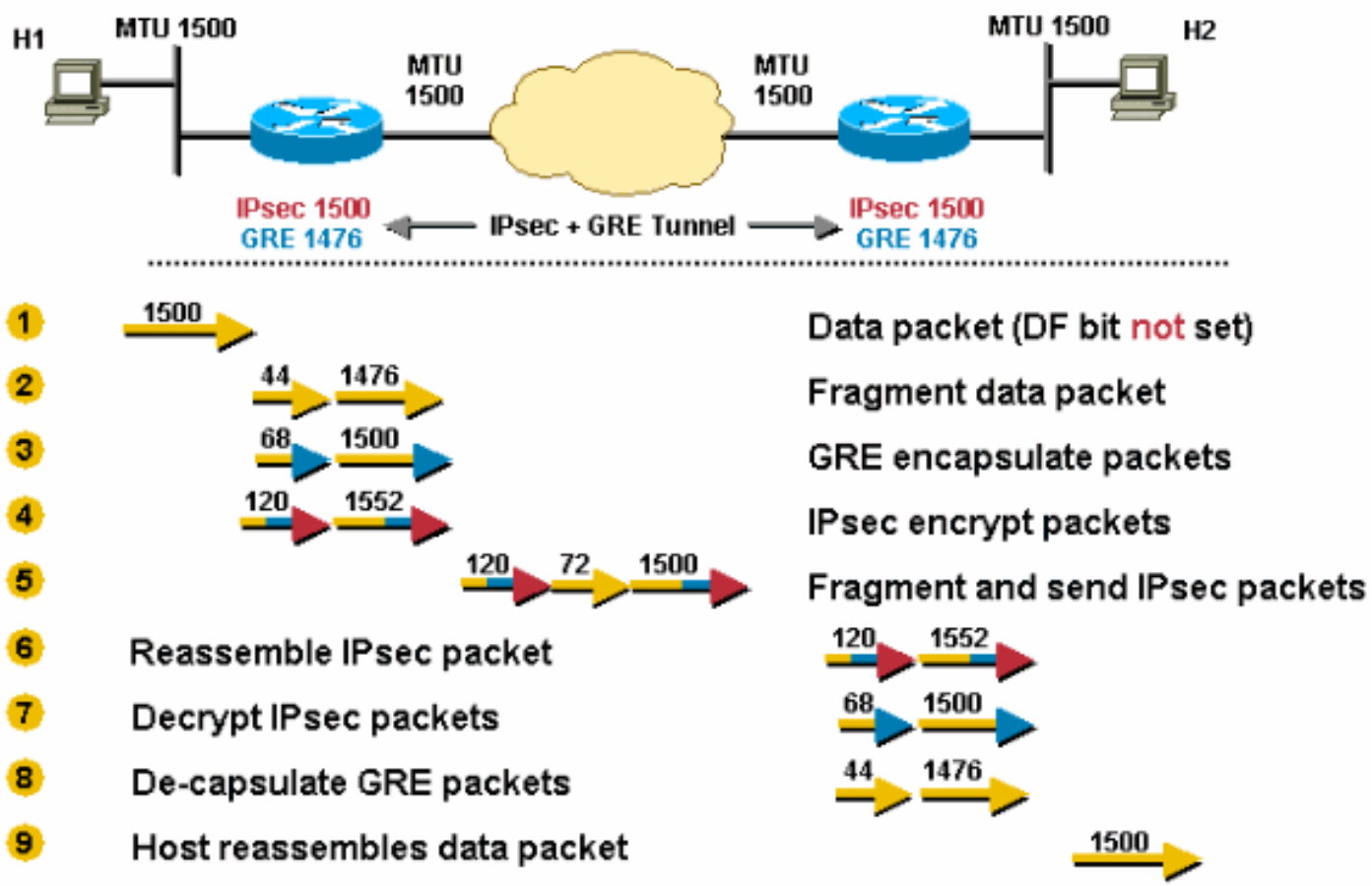
例 9

IPv4sec が GRE に加えて展開されています。発信物理 MTU は 1500、IPv4sec PMTU は 1500、そして GRE IPv4 MTU は 1476（1500 - 24 = 1476）です。

このため、TCP/IPv4 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 IPv4sec 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 IPv4sec 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」（IPv4sec トランスポート モード）または「ip mtu 1420」（IPv4sec 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。



- ルータは 1500 バイトのデータグラムを受信します。

- 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 IPv4 ヘッダー）の 2 つの断片にフラグメント化されます。

- GRE では IPv4 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 + IPv4sec パケットとなります。

- この 2 つのパケットは IPv4sec 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（IPv4sec トンネルモード）がそれぞれに追加されます。

- 1552 バイトの IPv4sec 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 IPv4 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、IPv4sec + GRE ピアに転送されます。

- 受信側ルータでは、2 つの IPv4sec フラグメント（1500 バイトと 72 バイト）が再構成されて、元の 1552 バイトの IPv4sec + GRE パケットが取得されます。120 バイトの IPv4sec + GRE パケットに対しては、必要な処理はありません。

- IPv4sec で 1552 バイトと 120 バイトの両方の IPv4sec + GRE パケットが復号され、1500 バイトと 68 バイトの GRE パケットが取得されます。

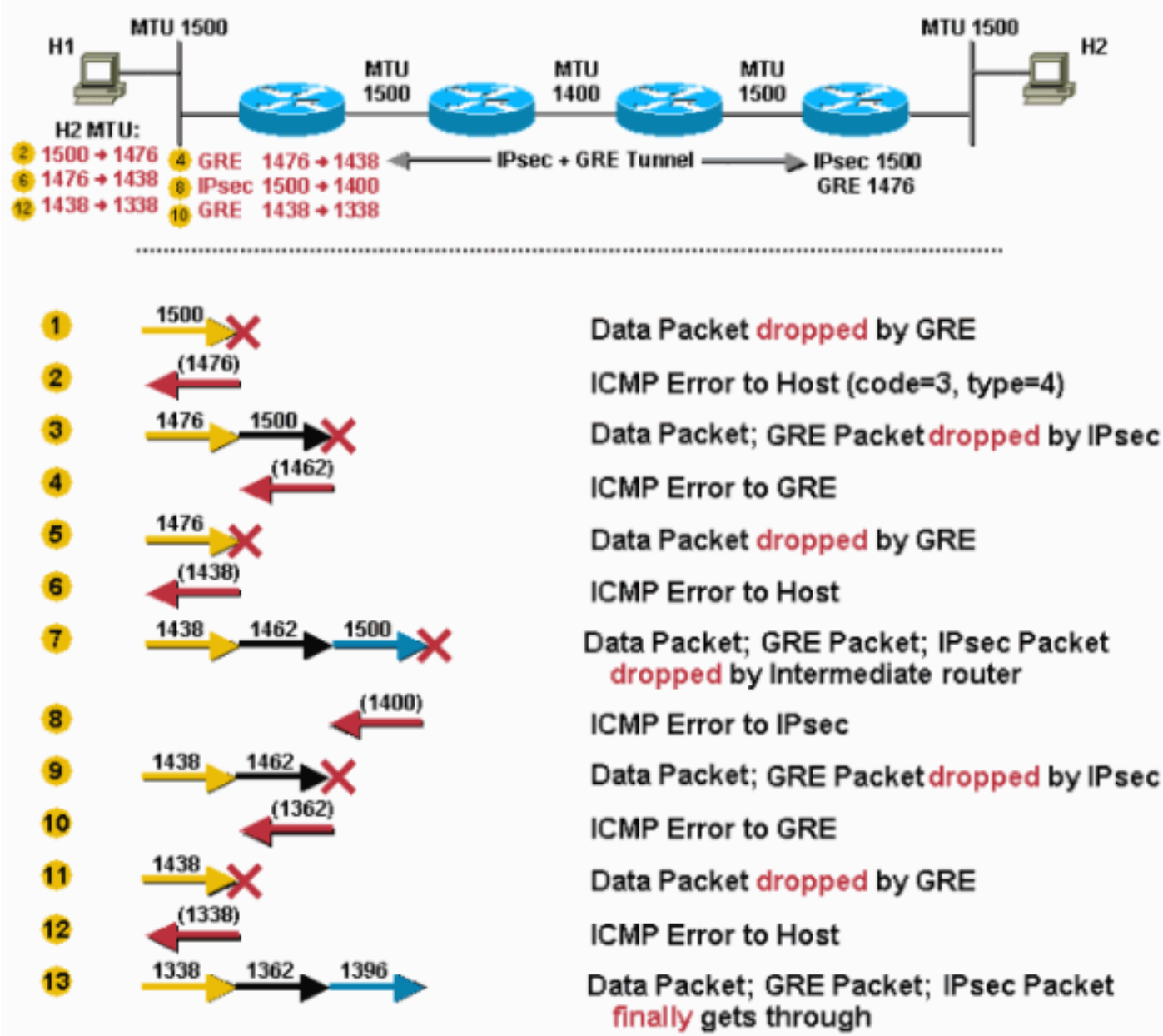
- GRE では、1500 バイトと 68 バイトの GRE パケットがカプセル化解除され、1476 バイトと 44 バイトの IPv4 パケット フラグメントが取得されます。これらの IPv4 パケット フラグメントが、宛先ホストに転送されます。

- Host 2 は、これらの IPv4 フラグメントを再構成して、元の 1500 バイトの IPv4 データグラムを取得します。

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。ホスト 1 と他のホスト間の（IPv4sec + 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/IPv4パケット上にDFビットが設定されます。

例 10



- 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。

- 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。

- 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 IPv4sec に渡されます。IPv4sec はパケットをドロップします。この理由は、GRE によって（設定状態の）DF ビットが内側の IPv4 ヘッダーからコピーされており、IPv4sec 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。

- IPv4sec 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 IPv4 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。

- 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。

- 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。

- 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 IPv4sec に渡します。

- IPv4sec 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、ドロップされます。

- 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 IPv4sec に送信します。この値は、関連する IPv4sec SA の PMTU 値内に IPv4sec により記録されます。

- 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 IPv4sec に渡します。IPv4sec はその PMTU を 1400 に変更しているので、このパケットをドロップします。

- IPv4sec 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。

- 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。

- 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。

- 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IPv4sec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 IPv4ヘッダー内のDFビットが常にクリアされます。

これにより、カプセル化されたデータ IPv4 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 IPv4 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た場合、次のようになります。

- GRE では、データ IPv4 ヘッダーから GRE IPv4 ヘッダーに、DF ビットをコピーします。
- GRE IPv4 ヘッダー内で DF ビットが設定されていると、IPv4sec 暗号化後のパケットが物理送信インターフェイスの IPv4 MTU に対して「大きすぎる」場合、IPv4sec はそのパケットをドロップし、GRE トンネルに IPv4 MTU サイズを縮小するように通知します。
- IPv4sec はそれ自体のパケットに対して PMTUD を実行しますが、IPv4sec PMTU が変更（縮小）されても、これは即座には、IPv4sec から GRE に通知されません。ところが、別のより大きなパケットが到着すると、ステップ 2 の処理が発生します。
- この場合、GRE の IPv4 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 IPv4 パケットをすべてドロップし、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IPv4 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IPv4 MTUに関連する、GREとIPv4sec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、IPv4secピア間のパスにもっと低いIPv4 MTUのリンクが存在する場合に、GREトンネルのIPv4 MTUをさらに低下させることができます。

GRE + IPv4sec 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、ここに示します。

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。
- ルータがTCP SYNパケットのTCP MSS値を低下させるように、トンネルインターフェイスで **ip tcp adjust-mss** コマンドを使用します。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 IPv4 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 IPv4 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 IPv4 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 IPv4sec 暗号化が実行され、物理送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、IPv4sec ピアでの IPv4 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
- 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 サポートとドキュメント - Cisco Systems

翻訳について
シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。