TCP の設定

TCP は、データ転送で使用される、データと確認応答のフォーマットを規定したプロトコルです。TCP は、通信の参加者がデータ転送の前に接続を確立する必要があるため、コネクション型のプロトコルです。フロー制御やエラー訂正を行うことで、TCP では順序通りにパケットが配送される信頼性が保証されます。IP パケット が損失する場合や順序通りに到達しない場合、TCP は正しいパケットを受信するまで再送を要求するので、信頼性があると見なされます。この章では、TCP に関する概念と、ネットワーク上での TCP の設定方法について説明します。

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュールで説明されるすべての機能がサポートされているとは限りません。最新の機能情報と注意事項については、ご使用のプラットフォームとソフトウェア リリースに対応したリリース ノートを参照してください。この章に記載されている機能の詳細、および各機能がサポートされているリリースのリストについては、「TCP の機能情報」（P.22）を参照してください。

プラットフォーム サポートとシスコ ソフトウェア イメージ サポートに関する情報を入手するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator には、http://www.cisco.com/go/cfn からアクセスします。Cisco.com のアカウントは必要ありません。

この章の構成

TCP の前提条件

TCP タイムスタンプ、TCP 選択的確認応答、および TCP ヘッダー圧縮

TCP タイムスタンプは送信と応答の双方で常に送られ、ヘッダーのタイムスタンプ値は常に変化するので、TCP ヘッダー圧縮では発信パケットを圧縮しません。シリアル リンクでの TCP ヘッダー圧縮を許可すると、TCP タイムスタンプ オプションはディセーブルにされます。シリアル ラインで TCP ヘッダー圧縮を使用する場合、TCP タイムスタンプと TCP 選択的確認応答はディセーブルにする必要があります。どちらの機能もデフォルトではディセーブルです。TCP 選択的確認応答がイネーブルの場合、ディセーブルにするには、**no ip tcp selective-ack** コマンドを使用します。

TCP の概要

TCP サービス

TCP は IP 環境で信頼性のあるデータ転送を提供します。TCP は Open Systems Interconnection（OSI）参照モデルのトランスポート層（レイヤ 4）に対応します。サービスの中で、TCP が提供するものとして、ストリーム データ転送、信頼性、能率的なフロー制御、全二重通信、およびデータ多重化があります。

ストリーム データ転送では、TCP はシーケンス番号で識別される構造化されないバイト ストリームを配送します。このサービスの利点は、アプリケーションがデータを TCP に渡す前にブロックに分ける必要がないことです。TCP は、バイト列をセグメント単位にグループ化し、IP に渡して配送させます。

TCP は、インターネットワークを介したエンドツーエンドの確実なパケット配送というコネクション型動作で信頼性を実現します。これは、受信側への確認応答で、発信側が次に受信を予期するバイト位置をその確認応答の番号として示し、バイト列を順序づけすることによって行います。指定された期間

に確認応答がないバイト列は再送されます。TCP の信頼性メカニズムを使用すれば、デバイスで、消失、遅延、重複、または破損したパケットを処理できます。タイムアウト メカニズムを使用すれば、デバイスで、消失パケットを検出して、再送信を要求できます。

TCP は効率的にフローを制御します。これは、受信 TCP プロセスが、送信元に確認応答を返すときに、内部バッファをオーバーフローさせずに受信可能な最も高いシーケンス番号を指定することを意味します。

TCP には全二重通信が備わっており、同時に送信と受信を処理できます。

TCP データ多重化では、同時に存在する多数の上位層の通信を、単一接続の上で多重化することができます。

TCP 接続の確立

信頼できる転送サービスを使用するには、TCP ホストは相手側とコネクション型のセッションを確立する必要があります。接続の確立は、「スリーウェイ ハンドシェイク」メカニズムを使用して実行されます。

スリーウェイ ハンドシェイクでは、接続の端点からの初期シーケンス値を両側で合意することで双方を同期します。また、このメカニズムでは、両側でデータ転送が可能になっていることと、お互いに相手側も転送が可能だと認識されることが保証されます。スリーウェイ ハンドシェイクは、セッションが確立されている間か終了した後で、パケットを転送しないため、または再送するために必要です。

各ホストは、送信しているストリーム内のバイト位置を追跡するために使われるシーケンス番号をランダムに選択します。その後、スリーウェイ ハンドシェイクは次のように進行します。

- 最初のホスト（ホスト A）が、初期シーケンス番号（X）と接続の要求を示すために同期開始（SYN）ビットを設定したパケットを送信して、接続を開始します。
- 2 番目のホスト（ホスト B）が SYN を受信し、シーケンス番号 X を記録し、SYN 確認応答（ACK = X + 1）によって応答します。ホスト B は自分自身の初期シーケンス番号（SEQ = Y）を含めます。ACK = 20 は、そのホストがバイト 0 ～ 19 を受信済みで、次はバイト 20 を予期していることを示します。このテクニックは前方確認応答と呼ばれます。
- ホスト A は、ホスト B が送信したすべてのバイトを受け取ったことに対し、ホスト A が次に予期する受信バイト位置（ACK = Y + 1）を示す前方確認応答でこれに応答します。次にデータ転送が始まります。

TCP 接続試行時間

Cisco IOS ソフトウェアが TCP 接続の確立に試行する待ち時間を設定できます。接続試行時間はホスト パラメータなので、デバイスを通過するトラフィックについてではなく、デバイスを起源とするトラフィックについてだけ関連するものです。TCP 接続試行時間を設定するには、グローバル コンフィギュレーション モードで **ip tcp synwait-time** コマンドを使用します。デフォルトは 30 秒です。

TCP 選択的確認応答

TCP 選択的確認応答機能は、1 つの TCP データ ウィンドウから複数のパケットが損失する場合のパフォーマンスを改善します。

この機能ができる前は、累積する確認応答から使用できる限られた情報で、TCP 送信者はラウンドトリップ時間に関する 1 つの損失パケットについてだけ知ることができました。積極的な送信者は、早い段階でパケットを再送信できますが、そのような再送信セグメントがすでに正常受信されている可能性があります。

TCP 選択的確認応答機能はパフォーマンスの改善に役立ちます。受信側の TCP ホストは送信側に選択的確認応答パケットを返し、送信側に受信済みのデータを知らせることができます。言い換えると、受信側はパケットを順序通りに受け取らなかったということを通知できます。送信側は、それで（最初の損失パケット以降すべてではなく）欠けているデータ セグメントだけを再送できます。

選択的確認応答の前に、TCP が 8 パケット ウィンドウのうちパケット 4 と 7 を損失すると、TCP はパケット 1、2、および 3 の確認応答だけ受信します。パケット 4 ～ 8 を再送信する必要があります。選択的確認応答を使うと、TCP はパケット 1、2、3、5、6、および 8 の確認応答を受け取ります。パケット 4 と 7 だけを再送信する必要があります。

TCP 選択的確認応答は 1 つの TCP ウィンドウ内で複数のパケットが損失したときだけ使われます。この機能がイネーブルでも使用しない場合、パフォーマンスに影響はありません。TCP 選択的確認応答をイネーブルにするには、グローバル コンフィギュレーション モードで **ip tcp selective-ack** コマンドを使用します。

TCP 選択的確認応答の詳細については、RFC 2018 を参照してください。

TCP タイムスタンプ

TCP タイムスタンプ オプションによって、TCP ラウンドトリップ時間の計測精度が向上します。タイムスタンプは送信と応答の双方で常に送信され、ヘッダーのタイムスタンプ値はいつも変化するため、TCP ヘッダー圧縮では発信パケットを圧縮しません。シリアル リンクでの TCP ヘッダー圧縮を許可すると、TCP タイムスタンプ オプションはディセーブルにされます。TCP タイムスタンプ オプションをイネーブルにするには、**ip tcp timestamp** コマンドを使用します。

TCP タイムスタンプの詳細については、RFC 1323 を参照してください。TCP ヘッダー圧縮の詳細については、『*Cisco IOS Quality of Service Solutions Configuration Guide*』の「Configuring TCP Header Compression」の章を参照してください。

TCP 最大リード サイズ

Telnet や rlogin で、TCP が入力キューから一度に読み込むことのできる最大の文字数は、デフォルトで非常に大きな値（32 ビット整数で正の最大値）です。TCP 最大リード サイズ値を変更するには、グローバル コンフィギュレーション モードで **ip tcp chunk-size** コマンドを使用します。

この値を変更することは推奨しません。

TCP PMTUD

PMTUD は TCP 接続 エンドポイント間のネットワーク帯域幅利用効率を最大化する方式で、RFC 1191 で説明されています。IP PMTUD を使用すると、ホストは経路上のさまざまなリンクで許容される Maximum Transmission Unit（MTU; 最大伝送ユニット）サイズの差異をダイナミックに検出し、対処できます。フラグメンテーションが必要（パケットが **interface** コンフィギュレーション コマンドを使用してインターフェイスに対して設定した MTU よりも大きい場合）なのに、「don't fragment」（DF）ビットがセットされているため、ルータがデータグラムを転送できない場合があります。中間ゲートウェイが、「Fragmentation needed and DF bit set」Internet Control Message Protocol（ICMP）メッセージを送信ホストに送信して、問題を警告します。この ICMP メッセージを受信すると、ホストは仮定のパス MTU を減らし、その結果として経路上の全リンクの最小パケットサイズに適した、より小さなパケットを送信します。

デフォルトでは、TCP PMTUD はディセーブルです。この機能がイネーブルかディセーブルかに関わらず、既存の接続は影響を受けません。

異なるサブネット上のシステム間でバルク データを移動するために TCP 接続を使用する場合、この機能をイネーブルにすることを推奨します。Remote Source-Route Bridging（RSRB; リモート ソースルート ブリッジング）を TCP カプセル化、Serial Tunnel（STUN; シリアル トンネル）、X.25 Remote Switching（XOT; X.25 リモート スイッチング、X.25 over TCP とも呼ばれます）、および何らかのプロトコル変換構成で使用している場合も、この機能をイネーブルにすることを推奨します。

ホストとして動作するルータが開設した接続への PMTUD をイネーブルにするには、**ip tcp path-mtu-discovery** グローバル コンフィギュレーション コマンドを使用します。

PMTUD の詳細については、『*Cisco IOS IP Application Services Configuration Guide*』の「Configuring IP Services」の章を参照してください。

TCP ウィンドウ スケーリング

TCP ウィンドウ スケーリング機能は、RFC 1323「*TCP Extensions for High Performance*」内のウィンドウ スケーリング オプションに対するサポートを追加します。Long Fat Network（LFN; 広帯域高遅延ネットワーク）と呼ばれる大きな帯域遅延積の特性を持つネットワーク経路での TCP のパフォーマンスを改善するため、より大きなウィンドウ サイズが推奨されます。TCP ウィンドウ スケーリングの強化で、そのサポートを提供します。

Cisco IOS ソフトウェアでのウィンドウ スケーリング拡張は TCP ウィンドウの定義を 32 ビットに拡大し、この 32 ビット値を TCP ヘッダーの 16 ビットウィンドウ フィールドに適合させるため、スケール係数を使用します。ウィンドウ サイズはスケール係数 14 まで大きくすることができます。典型的なアプリケーションは、広帯域高遅延ネットワークで動作するときにスケール係数 3 を使います。

TCP ウィンドウ スケーリング機能は RFC 1323 に準拠しています。最大ウィンドウ サイズは、1,073,741,823 バイトに増加しています。より大きなスケーラブル ウィンドウ サイズによって、広帯域高遅延ネットワーク上での TCP のパフォーマンスを向上できます。TCP ウィンドウ サイズを設定するには、グローバル コンフィギュレーション モードで **ip tcp window-size** コマンドを使用します。

TCP スライディング ウィンドウ

TCP スライディング ウィンドウにより、ホストは確認応答を待つ前に複数のバイト列やパケットを送信できるため、ネットワークの帯域幅をより効率的に使用できます。

TCP では、受信側は現在のウィンドウ サイズをすべてのパケットに設定します。TCP はバイト ストリーム 接続を提供しているため、ウィンドウ サイズはバイト単位で表現されます。ウィンドウは、送信者が確認応答を待機する前に送信できるデータ バイト数です。初期ウィンドウ サイズは接続確立時に示されますが、フロー制御によってデータ転送の間に変わる可能性があります。ウィンドウ サイズが 0 のときは「データ送信禁止」を意味します。デフォルトの TCP ウィンドウ サイズは 4128 バイトです。ルーターが大きなパケット（536 バイトよりも大きい）を送信していると確認できない限り、デフォルト値をそのまま使用することを推奨します。デフォルトのウィンドウ サイズを変更するには、**ip tcp window-size** コマンドを使用します。

たとえば、TCP スライディング ウィンドウの動作で、ウィンドウ サイズが 5 バイトの受信側に送るバイト シーケンス（1 ～ 10 の番号が付いた）があるとします。送信側は、最初の 5 バイトを取り囲むようにウィンドウを配置して、それをまとめて送信します。送信側はその後、確認応答を待ちます。

受信側は、ACK = 6 で応答します。これは 1 ～ 5 バイトを受け取り、次に 6 バイト目を予期していることを示します。同じパケットの中では、ウィンドウ サイズが 5 だと示します。送信側はスライディング ウィンドウを右に 5 バイト分ずらし、6 ～ 10 バイトを転送します。受信側は、ACK = 11 で応答します。これは、次に 11 バイト目を予期していることを示します。このパケットで、受信側はウィンドウ サイズが 0 であると示すことができます（例えば、内部バッファがいっぱいになったため）。この時点では、受信側からウィンドウ サイズが 1 以上の別パケットが送信されるまで、送信側はこれ以上のバイト列を送信できません。

TCP 発信キューサイズ

接続に TTY が関連付けられている場合（たとえば Telnet 接続など）、接続ごとの TCP 発信キューサイズはデフォルトで 5 セグメントです。接続に関連付けられている TTY がない場合、デフォルトのキューサイズは 20 セグメントです。デフォルト値を 5 セグメントから変更するには、**ip tcp queuemax** コマンドを使用します。

TCP 輻輳回避

TCP 輻輳回避機能を使用すると、単一のウィンドウ内で複数パケットが損失しているとき、TCP 送信側に対する確認応答パケットをモニタできます。以前は、送信側は高速リカバリ モードを終了するか、3 以上の重複確認応答パケットを待ってから次の未応答パケットを再送信するか、または再送タイマーのスロー スタートを待ちました。これは、パフォーマンスの問題になることがありました。

RFC 2581 および RFC 3782 の実装では、高速リカバリの期間に受信する部分確認応答への応答を組み込む高速リカバリ アルゴリズムの改良に対応し、単一のウィンドウ内で複数パケットが損失している状況でのパフォーマンスを改善します。

この機能は、既存の高速リカバリ アルゴリズムの強化です。この機能をイネーブルまたはディセーブルにするコマンドはありません。

debug ip tcp transactions コマンドの出力は、次の状態を表示することによって、確認応答パケットをモニタするように拡張されています。

- 高速リカバリ モードに移行した TCP。
- 高速リカバリ モードの間に受信した重複する確認応答。
- 受信した部分確認応答。

TCP 明示的輻輳通知

Explicit Congestion Notification（ECN; TCP 明示的輻輳通知）機能では、中間のルータが端点のホストにネットワーク輻輳が差し迫っていることを通知できるようになります。また、Telnet、Web 閲覧、音声や映像データの転送を含む、遅延やパケット損失の影響を受けるアプリケーションに関連付けられた TCP セッションのサポートも強化されています。この機能の利点は、データ転送時の遅延やパケット損失の軽減です。TCP 明示的輻輳通知をイネーブルにするには、グローバル コンフィギュレーション モードで **ip tcp ecn** コマンドを使用します。

TCP MSS 調整

TCP MSS 調整機能では、ルータを通過する一時的なパケット（特に SYN ビットが設定された TCP セグメント）の Maximum Segment Size（MSS; 最大セグメント サイズ）を設定することができるようになります。中間のルータで SYN パケットが切り捨てられないように最大セグメント サイズ値を指定するには、インターフェイス コンフィギュレーション モードで **ip tcp adjust-mss** コマンドを使用します。

ホスト（通常は PC）がサーバと TCP セッションを開始するときは、TCP SYN パケットの MSS オプション フィールドを使って IP セグメント サイズをネゴシエートします。MSS フィールドの値は、ホスト上の MTU 設定によって決まります。PC のデフォルト MSS 値は 1500 バイトです。

PPP over Ethernet（PPPoE）標準は、1,492 バイトのみの MTU をサポートします。ホストと PPPoE での MTU サイズの不一致は、ホストとサーバの間にあるルータで 1500 バイトのパケットが損失し、PPPoE を介した TCP セッションが終了する原因となる場合があります。たとえホストでパス MTU

（パス全体から正しい MTU を検出します）がイネーブルでも、パス MTU が機能するためにホストからリレーする必要がある ICMP エラー メッセージをシステム管理者がディセーブルにすることがあり、セッションが損失する場合があります。

ip tcp adjust-mss コマンドで TCP SYN パケットの MSS 値を調整すると、TCP セッション損失防止の役に立ちます。

ip tcp adjust-mss コマンドは、ルータを通過する TCP 接続に対してのみ有効です。

ほとんどの場合、**ip tcp adjust-mss** コマンドの *max-segment-size* 引数の最適値は 1,452 バイトです。この値に IP ヘッダーの 20 バイト、TCP ヘッダーの 20 バイト、および PPPoE ヘッダーの 8 バイトを足すと、イーサネット リンクの MTU サイズに適合する 1500 バイトのパケットになります。

設定手順については、「一時的な TCP SYN パケットに対する MSS 値、および MTU の設定」（P.10）を参照してください。

TCP アプリケーション フラグ拡張

TCP アプリケーション フラグ拡張機能によって、TCP アプリケーションに関する追加のフラグが表示可能になります。フラグには、ステータスやオプションという 2 種類のタイプがあります。ステータス フラグは、再送タイムアウト、アプリケーション クローズ、リスンの同期（SYNC）ハンドシェイクなど、TCP 接続のステータスを示します。追加のフラグは、VPN Routing and Forwarding（VRF; VPN ルーティングおよび転送）インスタンスが設定されているかどうか、ユーザが待機中かどうか、キープアライブ タイマーが動作中かどうかなどの設定オプションのステートを示します。TCP アプリケーション フラグを表示するには、**show tcp** コマンドを使用します。

TCP Show 拡張

TCP Show 拡張機能では、ホスト名形式ではなく、IP 形式でアドレスを表示したり、接続に関連付けられた VRF テーブルを表示したりする機能が導入されています。全エンドポイントのステータスを IP 形式のアドレス付きで表示するには、**show tcp brief numeric** コマンドを使用します。

ゼロフィールドと TCP パケット

Cisco IOS Release 15.0(1)M、12.2(33)XNE、12.2(33)SXI1、および 12.2(33)SRE と Cisco IOS XE Release 2.5 よりも前のリリースでは、ルータ上でゼロフィールド TCP パケットが受信されると、TCP パケット カウンタがインクリメントされました。

Cisco IOS Release 15.0(01)M、12.2(33)XNE、12.2(33)SXI1、および 12.2(33)SRE と Cisco IOS XE Release 2.5 以降のリリースでは、ルータ上でゼロフィールド TCP パケットが受信されると、TCP パケット カウンタがインクリメントされません。

show ip traffic コマンドが設定されているときに、ゼロフィールド TCP パケットが受信されると、それが TCP 統計情報フィールドの下に 0 として表示されます。**debug ip tcp packet** コマンドが設定されており、ゼロフィールド TCP パケットが受信された場合は、次のようなデバッグ メッセージが表示されます。

```
Jan 19 21:57:28.487: TCP: Alert! Received a segment with cleared flags 10.4.14.49
```

TCP MIB for RFC 4022 サポート

TCP MIB for RFC 4022 サポート機能で、RFC 4022「*Management Information Base for the Transmission Control Protocol (TCP)*」に対するサポートが導入されました。RFC 4022 は、TCP の管理容易性を向上させるための TCP MIB の増分変更です。

選択されたプラットフォーム、Cisco IOS リリース、およびフィーチャ セットの MIB を検索してダウンロードするには、次の URL にある Cisco MIB Locator を使用します。

http://www.cisco.com/go/mibs

TCP の設定方法

- 「TCP パフォーマンス パラメータの設定」（P.8）（任意）
- 「一時的な TCP SYN パケットに対する MSS 値、および MTU の設定」（P.10）（任意）
- 「TCP パフォーマンス パラメータの確認」（P.11）（任意）

TCP パフォーマンス パラメータの設定

前提条件

- ウィンドウ スケーリングをサポートするには、リンクの両側を設定する必要があります。設定しないと、最大ウィンドウ サイズとしてデフォルトの 65,535 バイトが適用されます。
- リモートピアとのスリーウェイ ハンドシェイク中に ECN の機能がネゴシエートされるため、ECN をサポートするには、リモート ピアが ECN 対応である必要があります。

手順の概要

1. **enable**
2. **configure terminal**
3. **ip tcp synwait-time** *seconds*
4. **ip tcp path-mtu-discovery** [**age-timer** {*minutes* | **infinite**}]
5. **ip tcp selective-ack**
6. **ip tcp timestamp**
7. **ip tcp chunk-size** *characters*
8. **ip tcp window-size** *bytes*
9. **ip tcp ecn**
10. **ip tcp queuemax** *packets*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Router> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configure terminal** 例： `Router# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip tcp synwait-time** *seconds* 例： `Router(config)# ip tcp synwait-time 60`	（任意）Cisco IOS ソフトウェアが TCP 接続の確立を試行する待ち時間を設定します。 • デフォルトは 30 秒です。
ステップ 4	**ip tcp path-mtu-discovery** [**age-timer** {*minutes* | **infinite**}] 例： `Router(config)# ip tcp path-mtu-discovery age-timer 11`	（任意）PMTUD をイネーブルにします。 • **age-timer**：TCP がより大きな MSS でパス MTU を再評価する分単位の時間間隔です。デフォルト値は 10 分です。最大で 30 分です。 • **infinite**：age timer をディセーブルにします。
ステップ 5	**ip tcp selective-ack** 例： `Router(config)# ip tcp selective-ack`	（任意）TCP 選択的確認応答をイネーブルにします。
ステップ 6	**ip tcp timestamp** 例： `Router(config)# ip tcp timestamp`	（任意）TCP タイムスタンプをイネーブルにします。

	コマンドまたはアクション	目的
ステップ 7	**ip tcp chunk-size** *characters* 例： `Router(config)# ip tcp chunk-size 64000`	（任意）Telnet や rlogin に対する TCP 最大リード サイズを設定します。 **（注）** この値を変更することは推奨しません。
ステップ 8	**ip tcp window-size** *bytes* 例： `Router(config)# ip tcp window-size 75000`	（任意）TCP ウィンドウ サイズを設定します。 • *bytes* 引数には 0 ～ 1073741823 の整数を設定できます。ウィンドウ スケーリングが LFN をサポートできるようにするには、TCP ウィンドウ サイズを 65535 より大きくする必要があります。ウィンドウ スケーリングが設定されていない場合、デフォルトのウィンドウ サイズは 4128 です。 **（注）** Cisco IOS Release 15.0(1)M 以降では、*bytes* 引数を 68 ～ 1073741823 の整数に設定できます。
ステップ 9	**ip tcp ecn** 例： `Router(config)# ip tcp ecn`	（任意）TCP の ECN をイネーブルにします。
ステップ 10	**ip tcp queuemax** *packets* 例： `Router(config)# ip tcp queuemax 10`	（任意）TCP 発信キュー サイズを設定します。

一時的な TCP SYN パケットに対する MSS 値、および MTU の設定

ルータを通過する一時的なパケット（特に SYN ビットが設定された TCP セグメント）の MSS を設定し、IP パケットの MTU サイズを設定するには、この作業を実行します。

ip tcp adjust-mss コマンドと同じインターフェイス上で **ip mtu** コマンドを設定する場合は、次のコマンドと値を使用することを推奨します。

- **ip tcp adjust-mss 1452**
- **ip mtu 1492**

手順の概要

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip tcp adjust-mss** *max-segment-size*
5. **ip mtu** *bytes*
6. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Router> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configure terminal** 例： `Router# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *type number* 例： `Router(config)# interface ethernet 0/1`	インターフェイス タイプを設定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**ip tcp adjust-mss** *max-segment-size* 例： `Router(config-if)# ip tcp adjust-mss 1452`	ルータを通過する TCP SYN パケットの MSS 値を調整します。 • *max-segment-size* 引数には、MSS をバイト単位で指定します。指定できる値の範囲は 500 ～ 1460 です。
ステップ 5	**ip mtu** *bytes* 例： `Router(config-if)# ip mtu 1492`	各インターフェイスにおいて送信される IP パケットの MTU サイズをバイト単位で設定します。
ステップ 6	**end** 例： `Router(config-if)# end`	グローバル コンフィギュレーション モードに戻ります。

TCP パフォーマンス パラメータの確認

手順の概要

1. **show tcp** [*line-number*] [**tcb** *address*]
2. **show tcp brief** [**all** | **numeric**]
3. **debug ip tcp transactions**
4. **debug ip tcp congestion**

手順の詳細

ステップ 1 **show tcp** [*line-number*] [**tcb** *address*]

TCP 接続のステータスを表示します。引数およびキーワードは次のとおりです。

- *line-number*：（任意）Telnet 接続ステータスの絶対行番号。
- **tcb**：（任意）ECN 対応の接続の Transmission Control Block（TCB; 転送制御ブロック）。
- *address*：（任意）TCB アドレス（16 進数）。有効な範囲は、0x0 ～ 0xFFFFFFFF です。

次に、ECN が利用可能な接続に関する詳細情報を 16 進数アドレスで表示する **show tcp tcb** コマンドの出力例を示します。

```
Router# show tcp tcb 0x62CD2BB8

Connection state is LISTEN, I/O status: 1, unread input bytes: 0
Connection is ECN enabled
Local host: 10.10.10.1, Local port: 179
Foreign host: 10.10.10.2, Foreign port: 12000

Enqueued packets for retransmit: 0, input: 0 mis-ordered: 0 (0 bytes)

Event Timers (current time is 0x4F31940):
Timer          Starts    Wakeups            Next
Retrans             0          0             0x0
TimeWait            0          0             0x0
AckHold             0          0             0x0
SendWnd             0          0             0x0
KeepAlive           0          0             0x0
GiveUp              0          0             0x0
PmtuAger            0          0             0x0
DeadWait            0          0             0x0

iss:           0 snduna:           0 sndnxt:           0     sndwnd:      0
irs:           0 rcvnxt:           0 rcvwnd:        4128  delrcvwnd:      0

SRTT: 0 ms, RTTO: 2000 ms, RTV: 2000 ms, KRTT: 0 ms
minRTT: 60000 ms, maxRTT: 0 ms, ACK hold: 200 ms
Flags: passive open, higher precedence, retransmission timeout

TCB is waiting for TCP Process (67)

Datagrams (max data segment is 516 bytes):
Rcvd: 6 (out of order: 0), with data: 0, total data bytes: 0
Sent: 0 (retransmit: 0, fastretransmit: 0), with data: 0, total data
bytes: 0
```

Cisco IOS ソフトウェア モジュラリティ

次に、ソフトウェア モジュラリティ イメージから **show tcp tcb** コマンドの出力例を示します。

```
Router# show tcp tcb 0x1059C10

Connection state is ESTAB, I/O status: 0, unread input bytes: 0
Local host: 10.4.2.32, Local port: 23
Foreign host: 10.4.2.39, Foreign port: 11000
VRF table id is: 0

Current send queue size: 0 (max 65536)
Current receive queue size: 0 (max 32768)  mis-ordered: 0 bytes

Event Timers (current time is 0xB9ACB9):
Timer          Starts    Wakeups            Next(msec)
```

```
Retrans             6          0                0
SendWnd             0          0                0
TimeWait            0          0                0
AckHold             8          4                0
KeepAlive          11          0          7199992
PmtuAger            0          0                0
GiveUp              0          0                0
Throttle            0          0                0

irs:    1633857851  rcvnxt: 1633857890  rcvadv: 1633890620  rcvwnd:  32730
iss:    4231531315  snduna: 4231531392  sndnxt: 4231531392  sndwnd:   4052
sndmax: 4231531392  sndcwnd:     10220

SRTT: 84 ms,  RTTO: 650 ms,  RTV: 69 ms,  KRTT: 0 ms
minRTT: 0 ms,  maxRTT: 200 ms, ACK hold: 200 ms

Keepalive time: 7200 sec, SYN wait time: 75 sec
Giveup time: 0 ms, Retransmission retries: 0, Retransmit forever: FALSE

State flags: none

Feature flags: Nagle

Request flags: none
Window scales: rcv 0, snd 0, request rcv 0, request snd 0
Timestamp option: recent 0, recent age 0, last ACK sent          0

Datagrams (in bytes): MSS 1460, peer MSS 1460, min MSS 1460, max MSS 1460
Rcvd: 14 (out of order: 0), with data: 10, total data bytes: 38
Sent: 10 (retransmit: 0, fastretransmit: 0), with data: 5, total data bytes: 76

Header prediction hit rate: 72 %

Socket states: SS_ISCONNECTED, SS_PRIV

Read buffer flags: SB_WAIT, SB_SEL, SB_DEL_WAKEUP
Read notifications: 4

Write buffer flags: SB_DEL_WAKEUP
Write notifications: 0
Socket status: 0
```

ステップ 2　**show tcp brief** [**all** | **numeric**]

（任意）アドレスを IP 形式で表示します。

TCP 接続のエンドポイントに関する簡潔な説明を表示するには、**show tcp brief** コマンドを使用します。Domain Name System（DNS; ドメイン ネーム システム）ホスト名形式のアドレスですべてのエンドポイントに関するステータスを表示するには、オプションの **all** キーワードを使用します。このキーワードを使用していない場合は、LISTEN ステートのエンドポイントは表示されません。IP 形式のアドレスですべてのエンドポイントに関するステータスを表示するには、オプションの **numeric** キーワードを使用します。

（注）　ルータで **ip domain-lookup** コマンドがイネーブルになっていて **show tcp brief** コマンドが実行された場合、出力表示のためのルータ応答時間は非常に遅くなります。応答時間を早くするには、**ip domain-lookup** コマンドをディセーブルにします。

次に、ユーザが Telnet でシステムに接続している間の **show tcp brief** コマンドでの出力例を示します。

```
Router# show tcp brief

TCB       Local Address           Foreign Address        (state)
609789AC  Router.cisco.com.23     cider.cisco.com.3733   ESTAB
```

次の例は、**numeric** キーワードを使用して IP 形式のアドレスが表示された後の IP アクティビティを示しています。

```
Router# show tcp brief numeric

TCB           Local Address          Foreign Address     (state)
6523A4FC      10.1.25.3.11000        10.1.25.3.23         ESTAB
65239A84      10.1.25.3.23           10.1.25.3.11000      ESTAB
653FCBBC      *.1723 *.* LISTEN
```

ステップ 3　debug ip tcp transactions

状態変化、再送、重複パケットのように重要な TCP トランザクションに関する情報を表示するには、**debug ip tcp transactions** コマンドを使用します。このコマンドは、データリンク レイヤ上層に分離される TCP/IP ネットワークでのパフォーマンス上の問題をデバッグするときに特に有用です。

次に、**debug ip tcp transactions** コマンドの出力例を示します。

```
Router# debug ip tcp transactions

TCP: sending SYN, seq 168108, ack 88655553
TCP0: Connection to 10.9.0.13:22530, advertising MSS 966
TCP0: state was LISTEN -> SYNRCVD [23 -> 10.9.0.13(22530)]
TCP0: state was SYNSENT -> SYNRCVD [23 -> 10.9.0.13(22530)]
TCP0: Connection to 10.9.0.13:22530, received MSS 956
TCP0: restart retransmission in 5996
TCP0: state was SYNRCVD -> ESTAB [23 -> 10.9.0.13(22530)]
TCP2: restart retransmission in 10689
TCP2: restart retransmission in 10641
TCP2: restart retransmission in 10633
TCP2: restart retransmission in 13384 -> 10.0.0.13(16151)]
TCP0: restart retransmission in 5996 [23 -> 10.0.0.13(16151)]
```

debug ip tcp transactions コマンド出力の次の行は、TCP が高速リカバリモードに移行したことを示しています。

```
fast re-transmit - sndcwnd - 512, snd_last - 33884268765
```

debug ip tcp transactions コマンド出力の次の行は、TCP が高速リカバリ モードのときに重複確認応答が受信されたこと（1 行目）と、部分確認応答が受信されていたこと（2 行目）を示しています。

```
TCP0:ignoring second congestion in same window sndcwn - 512, snd_1st - 33884268765
TCP0:partial ACK received sndcwnd:338842495
```

ステップ 4　debug ip tcp congestion

debug ip tcp congestion コマンドは、TCP 輻輳イベントに関する情報を表示するために使用します。また、**debug ip tcp congestion** コマンドは、データリンク層上で切り分けた、TCP/IP ネットワーク上の性能上の問題をデバッグするために使用できます。さらに、このコマンドは、TCP の送信ウィンドウ、輻輳ウィンドウ、および輻輳しきい値ウィンドウ内のばらつきに関する情報も表示します。

次に、**debug ip tcp congestion** コマンドの出力例を示します。

```
Router# debug ip tcp congestion

*May 20 22:49:49.091: Setting New Reno as congestion control algorithm
*May 22 05:21:47.281: Advance cwnd by 12
*May 22 05:21:47.281: TCP85FD0C10: sndcwnd: 1472
```

```
*May 22 05:21:47.285: Advance cwnd by 3
*May 22 05:21:47.285: TCP85FD0C10: sndcwnd: 1475
*May 22 05:21:47.285: Advance cwnd by 3
*May 22 05:21:47.285: TCP85FD0C10: sndcwnd: 1478
*May 22 05:21:47.285: Advance cwnd by 9
*May 22 05:21:47.285: TCP85FD0C10: sndcwnd: 1487
.
.
.
*May 20 22:50:32.559: [New Reno] sndcwnd: 8388480 ssthresh: 65535 snd_mark: 232322
*May 20 22:50:32.559: 10.168.10.10:42416 <---> 10.168.30.11:49100 congestion window
changes
*May 20 22:50:32.559: cwnd from 8388480 to 2514841, ssthresh from 65535 to 2514841
```

Cisco IOS TCP では、New Reno がデフォルト輻輳制御アルゴリズムです。ただし、アプリケーションで Binary Increase Congestion Control（BIC）を輻輳制御アルゴリズムとして使用することもできます。次に、BIC 輻輳制御アルゴリズムを使用した **debug ip tcp congestion** コマンドからの出力例を示します。

```
Router# debug ip tcp congestion

*May 22 05:21:42.281: Setting BIC as congestion control algorithm
*May 22 05:21:47.281: Advance cwnd by 12
*May 22 05:21:47.281: TCP85FD0C10: sndcwnd: 1472
*May 22 05:21:47.285: Advance cwnd by 3
*May 22 05:21:47.285: TCP85FD0C10: sndcwnd: 1475
*May 22 05:21:47.285: Advance cwnd by 3
*May 22 05:21:47.285: TCP85FD0C10: sndcwnd: 1478
*May 22 05:21:47.285: Advance cwnd by 9
*May 22 05:21:47.285: TCP85FD0C10: sndcwnd: 1487
.
.
.
.
.
.
*May 20 22:50:32.559: [BIC] sndcwnd: 8388480 ssthresh: 65535 bic_last_max_cwnd: 0
last_cwnd: 8388480
*May 20 22:50:32.559: 10.168.10.10:42416 <---> 10.168.30.11:49100 congestion window
changes
*May 20 22:50:32.559: cwnd from 8388480 to 2514841, ssthresh from 65535 to 2514841
*May 20 22:50:32.559: bic_last_max_cwnd changes from 0 to 8388480
```

TCP の設定例

例：TCP ECN の設定の確認

次の例では、TCP ECN が設定されていることを確認する方法を示します。

```
Router# show running-config

Building configuration...
.
.
.
ip tcp ecn ! ECN is configured.
.
.
.
```

次の例では、指定された接続（ローカル ホスト）上で、TCP が ECN 対応かどうかの確認方法を示します。

```
Router# show tcp tcb 123456A

!Local host
!
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Connection is ECN Enabled
Local host: 10.1.25.31, Local port: 11002
Foreign host: 10.1.25.34, Foreign port: 23
```

次の例では、1 つのアドレスについて簡易情報を表示させる方法を示しています。

```
Router# show tcp brief
!
TCB          Local address            Foreign Address        (state)
609789C      Router.cisco.com.23      cider.cisco.com.3733    ESTAB
```

次の例では、IP TCP ECN デバッグをイネーブルにする方法を示します。

```
Router# debug ip tcp ecn
!
TCP ECN debugging is on
!
Router# telnet 10.1.25.31

Trying 10.1.25.31 ...
!
01:43:19: 10.1.25.35:11000 <---> 10.1.25.31:23   out ECN-setup SYN
01:43:21: 10.1.25.35:11000 <---> 10.1.25.31:23   congestion window changes
01:43:21: cwnd from 1460 to 1460, ssthresh from 65535 to 2920
01:43:21: 10.1.25.35:11000 <---> 10.1.25.31:23   in non-ECN-setup SYN-ACK
```

TCP 接続が ECN を利用するときは、それ以前にホストは、Echo Congestion Experience（ECE; エコー輻輳経験）および Congestion Window Reduced（CWR; 輻輳ウィンドウ減少）ビットがヘッダーに設定されている ECN-setup SYN（synchronization）パケットをリモートの端点に送ります。ECE および CWR ビットを設定すると、輻輳のことではなく、送信中の TCP が ECN 対応であることをリモートの端点に示します。リモートの端点は、ECN-setup SYN-ACK（確認応答）パケットを送信側ホストに送ります。

この例の「out ECN-setup SYN」テキストは、ECE ビットと CWR ビットが設定された SYN パケットがリモート エンドに送信されたことを意味します。「in non-ECN-setup SYN-ACK」行は、リモートの端点は ECN 要求を承認する確認応答をしなかったため、このセッションでは ECN を利用できないことを示します。

次のデバッグ出力は、双方の端点で ECN 機能がイネーブルであることを示します。ECN-setup SYN に対し、相手側の端点が ECN-setup SYN-ACK メッセージを使用して承認の確認応答を返しました。この接続の以後のセッションでは、ECN を使用できます。

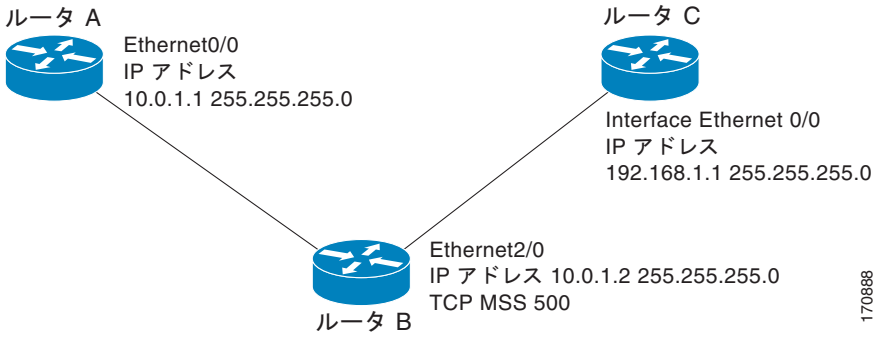
```
Router# telnet 10.10.10.10

Trying 10.10.10.10 ... Open
Password required, but none set
!
1d20h: 10.1.25.34:11003 <---> 10.1.25.35:23   out ECN-setup SYN
1d20h: 10.1.25.34:11003 <---> 10.1.25.35:23   in ECN-setup SYN-ACK
```

次は、ホストが接続されていることを確認する方法を示します。

```
Router# show debugging
!
TCP:
  TCP Packet debugging is on
  TCP ECN debugging is on
!
Router# telnet 10.1.25.234
!
Trying 10.1.25.234 ...
!
00:02:48: 10.1.25.31:11001 <---> 10.1.25.234:23   out ECN-setup SYN
00:02:48: tcp0: O CLOSED 10.1.25.234:11001 10.1.25.31:23 seq 1922220018
        OPTS 4 ECE CWR SYN  WIN 4128
00:02:50: 10.1.25.31:11001 <---> 10.1.25.234:23   congestion window changes
00:02:50: cwnd from 1460 to 1460, ssthresh from 65535 to 2920
00:02:50: tcp0: R SYNSENT 10.1.25.234:11001 10.1.25.31:23 seq 1922220018
        OPTS 4 ECE CWR SYN  WIN 4128
00:02:54: 10.1.25.31:11001 <---> 10.1.25.234:23   congestion window changes
00:02:54: cwnd from 1460 to 1460, ssthresh from 2920 to 2920
00:02:54: tcp0: R SYNSENT 10.1.25.234:11001 10.1.25.31:23 seq 1922220018
        OPTS 4 ECE CWR SYN  WIN 4128
00:03:02: 10.1.25.31:11001 <---> 10.1.25.234:23   congestion window changes
00:03:02: cwnd from 1460 to 1460, ssthresh from 2920 to 2920
00:03:02: tcp0: R SYNSENT 10.1.25.234:11001 10.1.25.31:23 seq 1922220018
        OPTS 4 ECE CWR SYN  WIN 4128
00:03:18: 10.1.25.31:11001 <---> 10.1.25.234:23   SYN with ECN disabled
00:03:18: 10.1.25.31:11001 <---> 10.1.25.234:23   congestion window changes
00:03:18: cwnd from 1460 to 1460, ssthresh from 2920 to 2920
00:03:18: tcp0: O SYNSENT 10.1.25.234:11001 10.1.25.31:23 seq 1922220018
        OPTS 4 SYN  WIN 4128
00:03:20: 10.1.25.31:11001 <---> 10.1.25.234:23   congestion window changes
00:03:20: cwnd from 1460 to 1460, ssthresh from 2920 to 2920
00:03:20: tcp0: R SYNSENT 10.1.25.234:11001 10.1.25.31:23 seq 1922220018
        OPTS 4 SYN  WIN 4128
00:03:24: 10.1.25.31:11001 <---> 10.1.25.234:23   congestion window changes
00:03:24: cwnd from 1460 to 1460, ssthresh from 2920 to 2920
00:03:24: tcp0: R SYNSENT 10.1.25.234:11001 10.1.25.31:23 seq 1922220018
        OPTS 4 SYN  WIN 4128
00:03:32: 10.1.25.31:11001 <---> 10.1.25.234:23   congestion window changes
00:03:32: cwnd from 1460 to 1460, ssthresh from 2920 to 2920
00:03:32: tcp0: R SYNSENT 10.1.25.234:11001 10.1.25.31:23 seq 1922220018
        OPTS 4 SYN  WIN 4128
    !Connection timed out; remote host not responding
```

例：TCP MSS 調整の設定

図 1　　TCP MSS 調整のトポロジ例



次の例では、図 1 に示すトポロジ例のインターフェイス調整値を設定して確認する方法を示します。ルータ B で、インターフェイスの調整値を設定します。

```
Router_B(config)# interface ethernet2/0
Router_B(config-if)# ip tcp adjust-mss 500
```

MSS 調整が設定されたルータ B を使用して、ルータ A からルータ C に Telnet します。

```
Router_A# telnet 192.168.1.1
Trying 192.168.1.1... Open
```

ルータ C からデバッグ出力を監視します。

```
Router_C# debug ip tcp transactions

Sep 5 18:42:46.247: TCP0: state was LISTEN -> SYNRCVD [23 -> 10.0.1.1(38437)]
Sep 5 18:42:46.247: TCP: tcb 32290C0 connection to 10.0.1.1:38437, peer MSS 500, MSS is
500
Sep 5 18:42:46.247: TCP: sending SYN, seq 580539401, ack 6015751
Sep 5 18:42:46.247: TCP0: Connection to 10.0.1.1:38437, advertising MSS 500
Sep 5 18:42:46.251: TCP0: state was SYNRCVD -> ESTAB [23 -> 10.0.1.1(38437)]
```

ルータ B で設定されたとおりに MSS が 500 に調整されます。

次の例は、MSS 値を 1452 にした PPPoE クライアントの設定を示します。

```
Router(config)# vpdn enable
Router(config)# no vpdn logging
Router(config)# vpdn-group 1
Router(config-vpdn)# request-dialin
Router(config-vpdn-req-in)# protocol pppoe
Router(config-vpdn-req-in)# exit
Router(config-vpdn)# exit
Router(config)# interface Ethernet0
Router(config-if)# ip address 192.168.100.1.255.255.255.0
Router(config-if)# ip tcp adjust-mss 1452
Router(config-if)# ip nat inside
Router(config-if)# exit
Router(config)# interface ATM0
Router(config-if)# no ip address
Router(config-if)# exit
Router(config-if)# no atm ilmi-keepalive
Router(config-if)# pvc 8/35
Router(config-if)# pppoe client dial-pool-number 1
Router(config-if)# dsl equipment-type CPE
Router(config-if)# dsl operating-mode GSHDSL symmetric annex B
```

```
Router(config-if)# dsl linerate AUTO
Router(config-if)# exit
Router(config)# interface Dialer1
Router(config-if)# ip address negotiated
Router(config-if)# ip mtu 1492
Router(config-if)# ip nat outside
Router(config-if)# encapsulation ppp
Router(config-if)# dialer pool 1
Router(config-if)# dialer-group 1
Router(config-if)# ppp authentication pap callin
Router(config-if)# ppp pap sent-username sohodyn password 7 141B1309000528
Router(config-if)# ip nat inside source list 101 Dialer1 overload
Router(config-if)# exit
Router(config)# ip route 0.0.0.0.0.0.0.0 Dialer1
Router(config)# access-list permit ip 192.168.100.0.0.0.0.255 any
```

例：TCP アプリケーション フラグ拡張の設定

次の出力は、**show tcp** コマンドを使用して表示されたフラグ（ステータスとオプション）を示しています。

```
Router# show tcp
.
.
.
Status Flags: passive open, active open, retransmission timeout
 App closed

Option Flags: vrf id set
IP Precedence value: 6
.
.
.
SRTT: 273 ms, RTTO: 490 ms, RTV: 217 ms, KRTT: 0 ms
minRTT: 0 ms, maxRTT: 300 ms, ACK hold: 200 ms
```

例：IP 形式でのアドレスの表示

次の例は、**numeric** キーワードを使用して IP 形式のアドレスを表示する IP アクティビティを示しています。

```
Router# show tcp brief numeric

TCB           Local Address          Foreign Address     (state)
6523A4FC      10.1.25.3.11000        10.1.25.3.23         ESTAB
65239A84      10.1.25.3.23           10.1.25.3.11000      ESTAB
653FCBBC      *.1723 *.* LISTEN
```

その他の参考資料

関連資料

内容	参照先
IP アドレッシングとサービス設定作業	『*Cisco IOS IP Addressing Services Configuration Guide*』
IP アプリケーション サービス コマンド：コマンド構文、コマンド モード、コマンド履歴、デフォルト設定、使用に関する注意事項、および例	『*Cisco IOS IP Application Services Command Reference*』
PMTUD	『*Configuring IP Services*』
TCP セキュリティ機能	• 『*TCP Out-of-Order Packet Support for Cisco IOS Firewall and Cisco IOS IPS*』 • 『*Configuring TCP Intercept (Preventing Denial-of-Service Attacks)*』
TCP ヘッダー圧縮、 クラスベースの TCP ヘッダー圧縮	• 『*Configuring Class-Based RTP and TCP Header Compression*』 • 『*Configuring TCP Header Compression*』
トラブルシューティング TCP	『*Internetwork Troubleshooting Handbook* 』の「*Troubleshooting TCP/IP*」の部分

規格

規格	タイトル
新しい規格または変更された規格はサポートされていません。また、既存の規格に対するサポートに変更はありません。	—

MIB

MIB	MIB リンク
CISCO-TCP-MIB	選択されたプラットフォーム、シスコ ソフトウェア リリース、およびフィーチャ セットの MIB を検索してダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

RFC

RFC	タイトル
RFC 793	「*Transmission Control Protocol*」
RFC 1191	「*Path MTU discovery*」
RFC 1323	「*TCP Extensions for High Performance*」
RFC 2018	「*TCP Selective Acknowledgment Options*」
RFC 2581	「*TCP Congestion Control*」
RFC 3168	「*The Addition of Explicit Congestion Notification (ECN) to IP*」
RFC 3782	「*The NewReno Modification to TCP's Fast Recovery Algorithm*」
RFC 4022	「*Management Information Base for the Transmission Control Protocol (TCP)*」

シスコのテクニカル サポート

説明	リンク
右の URL にアクセスして、シスコのテクニカル サポートを最大限に活用してください。 以下を含むさまざまな作業にこの Web サイトが役立ちます。 ・テクニカル サポートを受ける ・ソフトウェアをダウンロードする ・セキュリティの脆弱性を報告する、またはシスコ製品のセキュリティ問題に対する支援を受ける ・ツールおよびリソースへアクセスする - Product Alert の受信登録 - Field Notice の受信登録 - Bug Toolkit を使用した既知の問題の検索 ・Networking Professionals（NetPro）コミュニティで、技術関連のディスカッションに参加する ・トレーニング リソースへアクセスする ・TAC Case Collection ツールを使用して、ハードウェアや設定、パフォーマンスに関する一般的な問題をインタラクティブに特定および解決する この Web サイト上のツールにアクセスする際は、Cisco.com のログイン ID およびパスワードが必要です。	http://www.cisco.com/cisco/web/support/index.html

TCP の機能情報

表 1 に、この章に記載されている機能および具体的な設定情報へのリンクを示します。

プラットフォーム サポートとソフトウェア イメージ サポートに関する情報を入手するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator を使用すると、特定のソフトウェア リリース、フィーチャ セット、またはプラットフォームをサポートするソフトウェア イメージを確認できます。Cisco Feature Navigator には、http://www.cisco.com/go/cfn からアクセスします。Cisco.com のアカウントは必要ありません。

（注） 表 1 に、特定のソフトウェア リリース トレイン内の機能に対するサポートが導入されたソフトウェア リリースだけを示します。特に断りのないかぎり、そのソフトウェア リリース トレイン以降のリリースでもその機能がサポートされます。

表 1　TCP の機能情報

機能名	リリース	機能情報
TCP アプリケーション フラグ 拡張	12.4(2)T 12.2(31)SB2	TCP アプリケーション フラグ拡張機能によって、TCP アプリケーションに関する追加のフラグが表示可能になります。フラグには、ステータスやオプションという 2 種類のタイプがあります。ステータス フラグは、再送タイムアウト、アプリケーション クローズ、リスンの同期（SYNC）ハンドシェイクなど、TCP 接続のステータスを示します。追加のフラグは、VRF 識別が設定されているかどうか、ユーザが待機中かどうか、キープアライブ タイマーが動作中かどうかなどの設定オプションのステートを示します。 次の項では、この機能に関する情報について説明します。 • 「TCP アプリケーション フラグ拡張」（P.7） • 「TCP パフォーマンス パラメータの確認」（P.11） • 「例：TCP アプリケーション フラグ拡張の設定」（P.19） コマンド **show tcp** がこの機能により変更されました。

表 1　TCP の機能情報 （続き）

機能名	リリース	機能情報
TCP 輻輳回避	12.3(7)T	TCP 輻輳回避機能を使用すると、単一のウィンドウ内で複数パケットが損失しているとき、TCP 送信側に対する確認応答パケットをモニタできます。以前は、送信側は高速リカバリ モードを終了するか、3 以上の重複確認応答パケットを待ってから次の未応答パケットを再送信するか、または再送タイマーのスロー スタートを待ちました。これは、パフォーマンスの問題になることがありました。 RFC 2581 および RFC 3782 の実装では、高速リカバリの期間に受信する部分確認応答への応答を組み込む高速リカバリ アルゴリズムの改良に対応し、単一のウィンドウ内で複数パケットが損失している状況でのパフォーマンスを改善します。 この機能は、既存の高速リカバリ アルゴリズムの強化です。この機能をイネーブルまたはディセーブルにするコマンドはありません。 **debug ip tcp transactions** コマンドの出力は、次の状態を表示することによって、確認応答パケットをモニタするように拡張されています。 • 高速リカバリ モードに移行した TCP。 • 高速リカバリ モードの間に受信した重複する確認応答。 • 受信した部分確認応答。 この機能に関する詳細については、次の各項を参照してください。 • 「TCP 輻輳回避」（P.6） • 「TCP パフォーマンス パラメータの確認」（P.11） コマンド **debug ip tcp transactions** がこの機能により変更されました。

表 1　TCP の機能情報 （続き）

機能名	リリース	機能情報
TCP 明示的輻輳通知	12.3(7)T	Explicit Congestion Notification（ECN; TCP 明示的輻輳通知）機能では、中間のルータが端点のホストにネットワーク輻輳が差し迫っていることを通知できるようになります。また、Telnet、Web 閲覧、音声や映像データの転送を含む、遅延やパケット損失の影響を受けるアプリケーションに関連付けられた TCP セッションのサポートも強化されています。この機能の利点は、データ転送時の遅延やパケット損失の軽減です。 この機能に関する詳細については、次の各項を参照してください。 • 「TCP 明示的輻輳通知」（P.6） • 「TCP パフォーマンス パラメータの設定」（P.8） • 「TCP パフォーマンス パラメータの確認」（P.11） • 「例：TCP ECN の設定の確認」（P.16） **debug ip tcp ecn**、**ip tcp ecn**、**show debugging**、**show tcp** の各コマンドがこの機能により導入または変更されました。
TCP MIB for RFC 4022 サポート	Cisco IOS XE 3.1.0 SG 12.2(33)XN	TCP MIB for RFC 4022 サポート機能で、RFC 4022「*Management Information Base for the Transmission Control Protocol (TCP)*」に対するサポートが導入されました。RFC 4022 は、TCP の管理容易性を向上させるための TCP MIB の増分変更です。 この機能により、新規追加または変更されたコマンドはありません。
TCP MSS 調整	12.2(4)T 12.2(8)T 12.2(18)ZU2 12.2(28)SB 12.2(33)SRA 12.2(33)SXH 15.0(1)S	TCP MSS 調整機能では、ルータを通過する一時的なパケット（特に SYN ビットが設定された TCP セグメント）の Maximum Segment Size（MSS; 最大セグメント サイズ）を設定することができるようになります。 この機能は、12.2(4)T で初めて導入されました。 この機能により導入されたコマンドが、12.2(8)T で **ip adjust-mss** から **ip tcp adjust-mss** に変更されました。 12.2(28)SB および 12.2(33)SRA で、この機能がサブ インターフェイスで設定できるように強化されました。 この機能に関する詳細については、次の各項を参照してください。 • 「TCP MSS 調整」（P.6） • 「一時的な TCP SYN パケットに対する MSS 値、および MTU の設定」（P.10） • 「例：TCP MSS 調整の設定」（P.18） コマンド **ip tcp adjust-mss** がこの機能により導入されました。

表 1　TCP の機能情報 （続き）

機能名	リリース	機能情報
TCP Show 拡張	Cisco IOS XE 3.1.0 SG 12.4(2)T 12.2(31)SB2	TCP Show 拡張機能では、ホスト名形式ではなく、IP 形式でアドレスを表示したり、接続に関連付けられた VRF テーブルを表示したりする機能が導入されています。 次の項では、この機能に関する情報について説明します。 • 「TCP Show 拡張」（P.7） • 「TCP パフォーマンス パラメータの確認」（P.11） • 「例：IP 形式でのアドレスの表示」（P.19） コマンド **show tcp brief** がこの機能により変更されました。
TCP ウィンドウ スケーリング	12.2(8)T 12.2(31)SB2	TCP ウィンドウ スケーリング機能は、RFC 1323 のウィンドウ スケーリング オプションのサポートを追加しました。Long Fat Network（LFN; 広帯域高遅延ネットワーク）と呼ばれる広帯域で高遅延の特性を持つネットワーク経路での TCP のパフォーマンスを改善するため、より大きなウィンドウ サイズが推奨されます。TCP ウィンドウ スケーリングの強化で、そのサポートを提供します。 この機能に関する詳細については、次の各項を参照してください。 • 「TCP ウィンドウ スケーリング」（P.5） • 「TCP パフォーマンス パラメータの設定」（P.8） • 「TCP パフォーマンス パラメータの確認」（P.11） **ip broadcast-address** コマンドがこの機能で導入または変更されました。

用語集

LFN：Long Fat Network（広帯域高遅延ネットワーク）。高スループットで伝送距離が長い場合のネットワークで、帯域が広く遅延が大きいもの。衛星中継のネットワークは LFN の一例です。衛星リンクは伝播遅延が大きく、通常広い帯域幅を持ちます。

TCP：Transmission Control Protocol（伝送制御プロトコル）。信頼性のある全二重方式データ転送を提供するコネクション型のトランスポート レイヤ プロトコル。TCP は TCP/IP プロトコル スタックの一部です。